

Copyright

© 2016 PG Online Limited

The contents of this pack are protected by copyright.

This pack and all associated files distributed with it are supplied to you by PG Online Limited under licence and may be used and copied by you only in accordance with the terms of the licence agreement between you and PG Online Limited. Except as expressly permitted by the licence, no part of the materials distributed with this unit may be used, reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

Licence agreement

This is a legal agreement between you, the teaching institution, and PG Online Limited. PG Online Limited grants to you a non-exclusive, non-transferable, revocable licence to use this pack and all the associated files distributed with it in the course of teaching by your teachers and/or employees.

The materials distributed with this unit may be copied and used by your teachers and/or employees on a single site only in the course of their teaching. You warrant that you shall not, and shall procure that each of your teachers and/or employees shall not, share in any way any of the materials or part of the materials with any third party, including users on another site or individuals who are teachers and/or employees of a separate institution. You acknowledge and agree that the materials must remain with you, the teaching institution, and no part of the materials may be transferred to another institution. You also warrant that you shall not, and shall procure that each of your teachers and/or employees shall not, procure, authorise, encourage, facilitate or enable any third party to reproduce these materials in whole or in part without the prior permission of PG Online Limited.

In consideration of the licence granted to you, you shall indemnify PG Online Limited against all liabilities, costs, expenses, damages and losses (including but not limited to any direct, indirect or consequential losses, loss of profit, loss of reputation and all interest, penalties and legal costs and all other professional costs and expenses) suffered or incurred by PG Online Limited arising out of or in connection with the exercise by you of your rights granted under this licence.

Answers Pack

Introduction

This pack contains answers to in-text questions and exercises in the AQA AS and A Level Computer Science textbook by PM Heathcote and RSU Heathcote.

Suggested resources to accompany the sections in the textbook





To accompany each section in the textbook, there is a series of 12 teaching units for the AQA AS and A Level. Each unit contains editable PPT and DOC format materials to enable effective delivery of the content with relevant and engaging examples for students. There are worksheets and homework for each topic and an assessment text at the end of each unit with exam style questions. Answers to all worksheets, homework tasks and the assessment are also included.

Assessment

A few points to note:

This book contains several questions taken from past AQA exam papers and the answers are given in this guide. Other questions have been created for this textbook. We cannot predict or guarantee the areas covered in the sample questions will cover all areas that could come up in any given exam paper. That being said, when producing the questions, the following have been carefully taken into account:

* the range of questions is designed to elicit the understanding of students from E-A\* grade.
* appropriate command words and language is used across the range of questions (list, describe, state, discuss, explain…)
* questions worth between 1 and 5 marks, and an extended writing questions are provided, following the rough proportions of live exam papers.

Real exam papers go through a serious quality assurance process; feel free to use and adapt as you see fit.

Answers

Chapter 1 - Programming basics

# In-text questions

**Q1:** recipe: input = ingredients, output = pancakes, chocolate cake, etc.

knitting pattern: input = wool, output = scarf, sweater, etc.

set of directions: input = current location and destination, output = route.

**Q2:** In Python, you would need to use casting: billstring = str(billbetween3). Other languages will vary.

**Q3:** x = Come into the garden, Maud

y= 26

z= 23 (assuming counting starts at 1. In Python, array index starts at 0 so answer will be 22)

See Python program **Ch 1 Maud.py**

# Exercises

1. Suitable data type for each of the following data items:

Pupil’s surname **String**

A single letter indicating whether they are male or female **Char**

The amount owed for school trips **Real**

The number of school trips they have participated in **Integer**

Whether or not the pupil is entitled to free school meals **Boolean** [5]

2. (a) #program to split a bill between 3 people

OUTPUT "Enter total bill: "

bill 🡨 USERINPUT

OUTPUT "Enter number of people: "

n 🡨 USERINPUT

billbetween3 🡨 bill / 3

billbetween3 🡨round(billbetween3, 2)

roundstring 🡨 str(billbetween3)

OUTPUT billstring, roundstring

See Python program **Ch 1 bill** **between n.py** in the resources folder [6]

(b)

|  |  |  |  |
| --- | --- | --- | --- |
| Total bill | Number of people | Reason for test | Expected result |
| 23.65 | 4 | Test rounding – answer is 6.50375 – or any reasonable answer | 6.50 |
| xx | xx | Test invalid entry | Program will crash |

Any reasonable suggestions acceptable. [6]

3. (a) Answers include:  
Comments to say what sections of code do  
Meaningful variable names [2]

(b) # Program to count up takings from stall at fete

OUTPUT "Enter the number of 1p, 2p, 10p and 50p coins you have in the till"

OUTPUT “Number of 1p coins: "

onepCoins 🡨 USERINPUT

OUTPUT “Number of 2p coins: "

twopCoins 🡨 USERINPUT

OUTPUT “Number of 10p coins: "

tenpCoins 🡨 USERINPUT

OUTPUT “Number of 50p coins: "

fiftypCoins 🡨 USERINPUT

# calculate total in £

total 🡨 onepCoins/100 + twopCoins/50 + tenpCoins/10 + fiftypCoins/2

OUTPUT “Total value of coins = £”, total

See Python program **Ch 1 totalcoins.py** [6]

Chapter 2 - Selection

# In-text questions

**Q1:** In Python, for example, <> is written !=

**Q2:** Boolean

**Q3:** (depends on the programming language)

**Q4:** It finds the maximum of a, b and c

# Exercises

1.

(i) Membership: Premier Day: Weekday Time: 1700 **TRUE**

(ii) Membership: Adult Day: Weekday Time: 1100 **TRUE**

(iii) Membership: Junior Day: Weekday Time: 1000 **FALSE**

(iv) Membership: Adult Day: Weekend Time: 0900 **TRUE**

(v) Membership: Adult Day: Weekday Time: 1530 **FALSE** [5]

2. (a) #calculate cost of carpet

OUTPUT “Enter the longest dimension of room in metres: ”

length 🡨 USERINPUT

OUTPUT “Enter the shortest dimension of the room in metres: ”

width 🡨 USERINPUT IF width <= 4 THEN

IF length > 4 THEN

requiredLength 🡨 length

requiredWidth 🡨 4

ELSE

requiredLength 🡨 4

requiredWidth 🡨 width

ENDIF

carpetCost 🡨 (requiredLength\*requiredWidth) \* 10

OUTPUT "Length of carpet supplied = ", requiredLength

OUTPUT “Width of carpet supplied =", requiredWidth)

OUTPUT "Cost of carpet = £", carpetCost)

ELSE

OUTPUT "You must enter a width less than or equal to four metres"

ENDIF [5]

(b)

Length = 5, width = 3 **length = 5, width = 4, cost = £200**

Length = 5, width = 4 **length = 5, width = 4, cost = £200**

Length = 3, width = 2 **length = 4, width = 2, cost = £80**

Length = 3.9, width = 2 **length = 4, width = 2, cost = £80**

Length = 6, width = 5 **You must enter a width less than or equal to four** **metres** [5]

See Python program **Ch 2 carpet cost**

Chapter 3 – Iteration

# In-text questions

**Q1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **totalTemp** | **numberOfTemps** | **Temp<>-99** | **temp** | **averageTemp** |
| 0 | 0 | TRUE | 8 |  |
| 8 | 1 | TRUE | 12 |  |
| 20 | 2 | TRUE | **-100** |  |
| **-80** | **3** | **FALSE** |  | **-26.66667** |

**Q2:** The program will crash, because of division by zero.

The last two lines may be altered to

IF numberOfTemps = 0 THEN

OUTPUT ”No valid temperatures entered - program terminated”

ELSE

averageTemp 🡨 totalTemp / numberOfTemps

OUTPUT “average temp = ”, averageTemp

ENDIF

See **Python solutions to Section 1/Ch 3 average temperature**

**Q3:**  #program to test a user on the squares of numbers

#random(a,b) generates a random integer between a and b

anotherGo 🡨 “Y”

WHILE anotherGo = “Y”

num 🡨 random(1,25)

OUTPUT “What is the square of ”, numsquare

answer 🡨 USERINPUT()

numsquare 🡨 num \* num

IF answer = numsquare THEN

OUTPUT “correct, well done”

ELSE

OUTPUT “No, it is ”, numsquare

ENDIF

OUTPUT “Another go? Answer Y or N”

anotherGo 🡨 USERINPUT

ENDWHILE

Either loop is quite satisfactory here.

See Python program **Ch 3** **squares of numbers.py**.

**Q4:** Ans: “1 x 2 = 2”

**Q5:** The loop will end, and so will the program. ”yes” is not the same as ‘Y’ or ‘y’.

See Python program **Ch 3** **DieRoll.py**.

**Example 4:**

See Python program **Ch 3** **BlastOff.py**.

# Exercises

1. OUTPUT “Input highest number: “

highestNumber 🡨 USERINPUT

OUTPUT “Input multiplier: “

multiplier 🡨 USERINPUT

FOR i 🡨 2 TO highestNumber

result 🡨 i \* multiplier

OUTPUT result

ENDFOR

See Python program **Ch 3 multiplier.py**. Note that in Python the range has to be written:

*for i in range(2,highestNumber+1)*. [5]

2. import random

score 🡨 0

OUTPUT "which table would you like to be tested on? "

myTable 🡨 USERINPUT

FOR i 🡨 1 TO 5

testNum 🡨 random.randint (2,13)

rightAnswer 🡨 testNum \* myTable

OUTPUT myTable, " x", testNum, " = "

myAnswer 🡨 USERINPUT

IF myAnswer = rightAnswer THEN

OUTPUT "Correct! "

score 🡨 score + 1

ELSE

OUTPUT "Wrong... the correct answer is : ", rightAnswer

ENDIF

ENDFOR

OUTPUT "Your score out of 5 is ", score

See Python program **Ch 3** **Times table** **test.py**. [5]

Chapter 4 - Arrays

# In-text questions

**Q1:** 10

**Q2:** staff 🡨 ["Anna", "Bob", "Carol"]

quarterSales 🡨 [[100, 110, 120, 110],

[350, 355, 360, 360],

[200, 210, 220, 220]]

annualSales 🡨 0

FOR s 🡨 1 TO 3

#output staff name

***OUTPUT (“sales for ”, staff[s])***

FOR q 🡨 1 TO 4

OUTPUT "Quarter ", q, quarterSales[s][q]

annualSales 🡨 annualSales + quarterSales[s][q]

ENDFOR

ENDFOR

OUTPUT "Annual sales for all staff: ", annualSales

See Python program **Ch 4** **QuarterlySales.py**.

# Exercises

1. Referring to the BirdWatch program given earlier in this chapter:

(a) Because the loop always executes 8 times even if the bird was found on the first execution. [1]

(b) Rewrite the algorithm using a different type of loop.

birdFound 🡨 False

count 🡨 1

WHILE count < 9 and birdFound = False

IF bird = birdName[count]

birdFound 🡨 True

OUTPUT "number observed: "

birdsObserved 🡨 USERINPUT

birdCount[count] 🡨 birdCount[count] + birdsObserved

ENDIF

count 🡨 count + 1

ENDWHILE

See Python program **Ch 4** **birdwatch2.py**. [3]

2. babyWeight 🡨 [1700, 2030, 4400, 2900,1600, 3000, 3500,1560, 2800, 4600]

numBabies 🡨 len(babyWeight)

totalWeight 🡨 0

FOR i 🡨 1 TO 10

totalWeight 🡨 totalWeight + babyWeight[i]

ENDFOR

averageWeight 🡨 totalWeight / numBabies

OUTPUT “Average baby weight ”, averageWeight

totalWeight 🡨 0

numUnderweight = 0

FOR i 🡨 1 to 10

IF babyWeight[i] < averageWeight - 500

totalWeight 🡨 totalWeight + babyWeight[i]

numUnderweight 🡨 numUnderweight +1

ENDIF

ENDFOR

averageUnderweight 🡨 totalWeight/numUnderweight

OUTPUT "Average underweight ", averageUnderweight

OUTPUT "Number of underweight babies ", numUnderweight

See Python program **Ch 4** **baby weights.py**. [5]

3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mark 1 | Mark 2 | Mark 3 |
| Student 1 | 10 | 10 | 10 |
| Student 2 | 7 | 6 | 9 |
| Student 3 | 0 | 0 | 5 |
| Student 4 | 0 | 0 | 0 |
| Student 5 | 2 | 2 | 2 |

The test data does not include invalid data, such as non-numeric data - this could be handled using an exception-handling routine, covered in Chapter 6. However, students may include invalid data in the table and that is obviously good practice! [2]

mark = [[10,10,10],

[7,6,9],

[0,0,5],

[0,0,0],

[2,2,2]]

classTotal 🡨 0

FOR s 🡨 1 TO 5

stotal 🡨 0

FOR m 🡨1 TO 3

stotal 🡨 stotal + mark[s][m]

ENDFOR

IF stotal > 0 THEN

studentAverage 🡨 stotal/3

ELSE

studentAverage 🡨 0

ENDIF

OUTPUT "Average mark for student ",s, studentAverage

classTotal 🡨 classTotal + stotal

ENDFOR

classAvg 🡨 classTotal / 15

OUTPUT " Class average = ", classAvg

See Python program **Ch 4** **average** **class marks.py**. [4]

4. The code finds and outputs the coordinates of the hidden treasure.

import random

FOR x 🡨 0 TO 9

FOR y 🡨 0 TO 9

grid[x][y] 🡨 0

ENDFOR

ENDFOR

x 🡨 random.randint(0,9) #random integer in range 0..9

y 🡨 random.randint(0,9)

grid[x][y] 🡨 1 [5]

See Python program **Ch 4** **Treasure grid.py**.

Chapter 5 - Subroutines

# In-text questions

**Q1:** Students may list str, print, input, randint, float etc.

**Q2:** 5

**Q3:**

SUB getresponse

OUTPUT "Do you like chocolate? "

response 🡨 USERINPUT

WHILE response <> "y" AND response <> "n"

OUTPUT "Please answer y or n... " OUTPUT “Do you like chocolate? ”

response 🡨 USERINPUT ENDWHILE

IF response = "y” THEN

OUTPUT “Me too!”

ELSE

OUTPUT “You're kidding!”

ENDIF

ENDSUB

#main program calls the subroutine

getresponse

See Python program **Ch 5** **chocolate.py**.

**Q4:** 7 8 9 10 11 2 3 4 12

Note that lines 1 and 5 are “non-executable”.

**Q5:**

SUB addNumbers(n,m)

total 🡨 0

FOR count 🡨 n TO m

total 🡨 total + count

ENDFOR

RETURN total

ENDSUB

#main program

result 🡨 addNumbers(5,11)

OUTPUT “Sum = ”, result

See Python program **Ch 5** **add numbers.py**

**Q6:**

SUB assignGrade(mark):

IF mark >= 80

grade 🡨 "Distinction"

ELSE IF mark >= 65

grade 🡨 "Merit"

ELSE IF mark >= 50

grade 🡨 "Pass"

ELSE

grade 🡨 "Fail"

ENDIF

RETURN grade

ENDSUB

#main program

OUTPUT “Please enter student name: ”

name 🡨 USERINPUT

OUTPUT “Please enter mark ”

studentMark 🡨 USERINPUT

studentGrade 🡨 assignGrade(studentMark)

OUTPUT name, studentGrade

See Python program **Ch 5 student grade.py**

**Q7:** pi is a local variable. The program will return an error if you try to print it in the main program. Its ‘scope’ is the subroutine.

**Q8:** In example 4: 8 9 10 11 12 13 2 3 4 5 14

**Q9:** In the main program, a,b,c and x have values 4 5 6 10

In the subroutine, a,b,c and x have values 1 2 3 10

In the main program, a,b,c and x now have values 4 5 6 10

**Q10:** In sub printnum, y = 20

program will return an error because y is not defined in the main program

**Q11:** Answer in text

# Exercises

1. (a) Global variable: option in main program. All other variables in the subprograms are local.

local variable: choice in sub menuChoice, score1, score2, totalScore in sub playGame, several in playerTurn [4]

(b) Parameters player1, score1 are passed from sub playGame to sub playerTurn, where they are named player, score.

The advantage of using parameters is that it keeps the subroutine self-contained with no danger of using a variable name that has been used for some other purpose elsewhere in the program, and changing its value. [3]

(c) It calls the subroutine menuChoice and assigns the return value to the variable option. [2]

Chapter 6 - Files and exception handling

# In-text questions

**Q1:**

OPEN birdFile to append data

birdName 🡨 USERINPUT “Enter bird name, x to end: ”

WHILE birdName <> "x"

birdsSeen 🡨 USERINPUT “ total number of birds seen: ”

WRITELINE (birdFile, birdName,",",birdsReported)

birdName 🡨 input("Enter bird name, x to end: ")

ENDWHILE

CLOSE birdFile

**Q2:** It finds the total of all the birds in the file.

**Q3**: (i) Please try again… enter a number between 1 and 3:

(ii) "That is not an integer!”

Please try again… enter a number between 1 and 3:

# Exercises

1. OPEN scoreFile for reading

READ (scorefile, recordscore)

CLOSE scorefile

myscore = USERINPUT “input score: ”

IF myscore > recordscore THEN

OPEN scoreFile for writing

WRITE (scoreFile, recordscore)

CLOSE scoreFile

ENDIF

See Python program **Ch 6** **write record score.py** [5]

2. (a) Exception handling is used to stop the program crashing in the event of something occurring that would normally cause it to crash. For example reading past the end of file or trying to convert a string containing non-numeric characters to an integer

[2]

(b) They are useful because they can check for an invalid entry, e.g. text instead of numeric, and execute a statement asking the user to re-enter data, rather than just crashing when an attempt is made to convert the entry to an integer. [3]

Chapter 7 - Solving logic problems

# In-text questions

# Solutions to questions in textbook on page 37-38.

# Exercises (These exercises have not been awarded marks.)

1. Three days ago, yesterday was Saturday (the day before Sunday). So three days ago was Sunday. So tomorrow is Thursday.

2. The two wives W1 and W2 cross, W1 returns and the two husbands H1 and H2 cross. H1 returns and takes W1 across. (There are other solutions such as the two husbands going across first, and W2 returning instead of W1.)

3. The sequence of steps below shows the solution.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step number** | **8-pint jug** | **5-pint jug** | **3-pint jug** |
|  | 8 | 0 | 0 |
| 1 | 3 | 5 | 0 |
| 2 | 3 | 2 | 3 |
| 3 | 6 | 2 | 0 |
| 4 | 6 | 0 | 2 |
| 5 | 1 | 5 | 2 |
| 6 | 1 | 4 | 3 |

4. It can be done in 17 minutes. (Most people will get 19 minutes by making the fastest person return with the torch each time.)

Name the people 1, 2, 5 and 10 after the times they takes.

1 and 2 go across first (2 minutes)

2 returns (4 minutes)

5 and 10 go across (14 minutes)

1 returns (15 minutes)

1 and 2 go across (17 minutes)

Chapter 8 – Structured programming

# In-text questions

**Q1:** Global variables can be changed within a subroutine and this may not be noticed by a programmer who is updating the program. It is important to keep each subprogram self-contained.

**Q2:** Hierarchy chart for the program given in Q11 in Chapter 5, page 26.

dice game

display rules

display menu and validate choice

Initialise scores

get player names

players take turns

display winner

play game

roll dice

calculate player’s score

# Exercises

1. (i) It means that the subroutine is independent of the subroutine and changing the name of a variable in the main program will not affect the subroutine. Changing the value of a global variable will likewise not affect the subroutine. It is easier to trace where the value of a variable is set and where it changes. [3]

(ii) Use comments to explain what the sections of the program do, use meaningful variable names, split the program up into subroutines each with a well-defined task.

[6]

2.

[6]

Quiz

next question

initialise score

generate random number

read the question and answer file

display question and get answer

update score

ouput score

Chapter 9 - Writing and interpreting algorithms

# In-text questions

**Q1:** Governments, security agencies such as MI5 and the CIA, banks, e-commerce etc.

**Q2:**

1 n 🡨 0 ;initialise n

2 nsquared 🡨 n\*n

3 Is nsquared = xsquared?

4 If yes, output n. If no, add 1 to n and repeat from step 2

**Written as an algorithm using programming structures:**

n 🡨 0 ;initialise n

WHILE nsquared <> xsquared

n 🡨 n+1

nsquared 🡨 n\*n

ENDWHILE

OUTPUT n

**Q3:** It is not an efficient algorithm, executing in as few steps as possible.

xsquared 🡨 19321

low 🡨 1

high 🡨 xsquared

guess 🡨 int((low + high)/2)

nsquared 🡨 guess\*\*2 BLOCK 1 SEQUENCE

numberOfGuesses 🡨 1

WHILE nsquared <> xsquared

IF nsquared > xsquared

high 🡨 int((high + low)/2)

ELSE

low 🡨 guess

numberOfGuesses 🡨 numberOfGuesses + 1

ENDIF

guess 🡨 int((low + high)/2)

nsquared 🡨 guess\*\*2 BLOCK 2 - ITERATION

ENDWHILE

print(“square root is ”,guess) BLOCK 3 – SEQUENCE

OUTPUT numberOfGuesses

**Q4:** See coded statements in red above

See Python program **Ch 9** **square root binary search.py in Python solutions Section 2 folder**

In answer the question, “Is there a formula for calculating how many guesses it should take to find the square root?”, you could get students to carry out a series of tests on different numbers using the program **Ch 9** **square root binary search v2.py**

The answer appears to be “No”! Let us know if a bright student finds one!

612 = 3721, number of guesses 12

622 = 3844, number of guesses 11

632 = 3969, number of guesses 6

642 = 4096, number of guesses 6

652 = 4225, number of guesses 12

662 = 4356, number of guesses 11

**Q5:** You need a third variable to act as a temporary store temp. Move a[1] into the temporary store, then move the a[2] into a[1], the move temp into a[2]. (Think of swapping the contents of two fish tanks … you need a third tank to hold one lot temporarily!)

**Q6:** You could check on each pass whether any swaps had been made – if not, the items are sorted and you could exit the loop. This is not good programming practice as a block should have a single exit point, so it should be rewritten with a WHILE loop.

**Q7:** kl,mr!

See Python program **Ch 9** **Caesar cipher.py**

# Exercises

1. (a) Sequence: a series of two or more statements one after the other

Selection: e.g. IF…THEN…ELSE. A statement that selects the next statement to be performed based on whether a certain condition is true or false.

Iteration: A number of statements are repeated until a given condition is met.

[6]

(b) (i) Sequence [1]

(ii) 20 20 [1]

2.

IF TeamAGoals > TeamBGoals THEN

TeamAPoints 🡨 TeamAPoints + 3

ELSE

IF TeamBGoals > TeamAGoals THEN

TeamBPoints 🡨 TeamBPoints + 3

ENDIF

ENDIF

(There are other solutions – it is not necessary to use "ELSE".)

[3]

3. (a) (i) 5 1 6 5 1 6 … is a sequence in which 5 is one less than the previous number

[1]

(ii) Total of numbers not divisible by 3 [1]

(b) pseudocode:

OUTPUT "Enter 3 numbers"

num1 🡨 USERINPUT

num2 🡨 USERINPUT

num3 🡨 USERINPUT

valid = TRUE

total 🡨 num1 + num2 + num3

remainder 🡨 total MOD 3 ;MOD operator calculates remainder IF remainder <> 0 THEN

valid 🡨 FALSE

ELSE

IF num1 - num2 = 1 THEN

valid 🡨 FALSE

ELSE

IF num2 – num3 = 1 THEN

valid 🡨 FALSE

ELSE

IF num3 – num1 = 1 THEN

valid 🡨 FALSE

ENDIF

ENDIF

ENDIF

ENDIF

IF valid = FALSE THEN

OUTPUT "INVALID PATTERN"

ENDIF [7]

Chapter 10 - Testing

# In-text questions

**Q1:** Devise a test plan for the program in Q7, page 46.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test number** | **Test data** | **Purpose of test** | **Expected result** | **Actual result** |
| 1 | ABC … XYZ | Tests conversion of uppercase to lowercase, some non-alpha characters | The encoded message is: def … abc |  |
| 2 | Press Enter | Test what happens when no message entered | The encoded message is: |  |
| 3 | 123Aa!?z | Mixture of characters, numeric and punctuation | The encoded message is: 123dd!?c |  |
| 4 | Or similar tests |  |  |  |
| 5 |  |  |  |  |

# Exercises

1.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **x** | **y** | **z** |
|  | 0 | 0 | 0 |
| 5 | 5 | 1 | 0 |
| 7 | 12 | 2 | 0 |
| 2 | 14 | 3 | 0 |
| 2 | 16 | 4 | 0 |
| 4 | 20 | 5 | 0 |
| -1 | 20 | 5 | 4 |
|  |  |  |  |
|  |  |  |  |

[5]

2. An algorithm is a sequence of steps; specifying how to solve a problem. [2]

|  |  |  |
| --- | --- | --- |
| **Answer** | **Count** | **Remainder** |
| True | - | - |
|  | 2 | 1 |
|  | 3 | 1 |
|  | 4 | 3 |
|  | 5 | 2 |
|  | 6 | 1 |

[6]

Works out if x is a prime number *or* checks if x is divisible (with no remainder) [1]

Chapter 11 - Abstraction and automation

# In-text questions

**Q1:** The 50th number is 502 = 2,500

**Q2:** random, int, char, input, print, etc.

**Q3:** There are only 2 ways to solve the problem, clockwise or anti-clockwise. 1 has to go to 9, 3 has to go to 7, etc. Each of these is 4 moves, so 16 moves in total.

# Exercises

1. The program will ignore all details about the route except the roads, type of road, distances between intersections to calculate a route and the mileage. To calculate the journey time it will ignore the type of vehicle and use a calculated average time according to some algorithms. Time of day may be used in estimating current traffic conditions but most details will be ignored.

The map produced by the computer will show the route and general direction but not bridges, vehicles on the roads, buildings beside the road etc. [5]

2. In an unsophisticated implementation the cave could be represented simply by an outline shape and grid points, ignoring any other details. The movement of waves, the slipperiness of the rocks, the amount of light entering the cave, etc., may be irrelevant and are ignored. This is what is meant by information hiding. [5]

Procedural abstraction is used in the top down design of the program, where the subtasks are identified and written as procedures, called in a hierarchical structure using parameter passing. Then, if one aspect of the game is changed, only certain procedures will be affected. [5]

3. They use **procedural** abstraction in top down design, passing parameters to subroutines

**Functional** abstraction when calling functions such as random() to generate a random number, or user-written functions

**Data** abstraction when using data types such as int, real, string, etc, without knowing how these are actually represented [6]

4. Unfold the graph so it looks something like this:

4

1

7

6

3

2

5

8

Place the first coin at 1 and move it to 6. Then place a coin at 4 and move it to 1, etc. The maximum number of coins that can be placed is 7, since the 8th coin cannot be moved to an empty point. [1]

Chapter 12 – Finite state machines

# In-text questions

**Q1:** (i) and (iii) are valid

**Q2:** A valid string starts with one or more a’s followed by one or more b’s. (n.b. not “any number of a’s followed by any number of b’s” since this would include zero a’s and zero b’s, neither of which is allowed.)

# Exercises

1.

|  |  |  |
| --- | --- | --- |
| **Original state** | **Input** | **New state** |
| S0 | 10 | S10 |
| S0 | 20 | S20 |
| S0 | 50 | S50 |
| S0 | R | S0 |

[3]

20 ,20, 10

R, R 50

10, 20, 20

20, 50, 50

20, R, 50 [4]

2. (a)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Current state** | S1 | S1 | S2 | S2 | S3 | S3 | S4 | S4 | S5 | S5 |
| **Input symbol** | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| **Next state** | S2 | S3 | S2 | S4 | S3 | S3 | S4 | S5 | S5 | S4 |

[2]

(b) Accept/Accepting/Accepted (state) // Input (string) is accepted

**Accept** if the FSA finishes in this state output is Yes

**Reject** Stop state [2]

(c)

|  |  |
| --- | --- |
| **Input string** | **String accepted? (Yes/No)** |
| 101 | No |
| 000 | No |
| 010001101 | No |
| 0100011011 | Yes |

[2]

(d) Strings that start with a 0 [2]

Followed by any sequence containing an odd number of 1s and zero or more 0s;

3. (a) Lift stationary at floor X / stationary at floor Y (MAX 2) ;

moving (or equivalent) up ; moving down ;

full ;

out of order ;

emergency stopped;

door open ; door closed ;

lift has been requested for floor X (MAX 1) ;

**A.** 'stationary' for 1 (but not in addition to the above)

**A.** 'moving' alone for 1 (but not in addition to the above)

A. anything plausible

R. anything which reads like an action/input [3]

(b)

|  |  |  |
| --- | --- | --- |
| **Input** | **Current state** | **Next state** |
| 0 | S1 | S2 |
| 0 | S2 | S1 |
| 1 | S1 | S1 |
| 1 | S2 | S2 |

*Table 1*

[3]

Chapter 13 – Number systems

# In-text questions

**Q1**: 57 and 255

**Q2**: 00100101 and 01100100

**Q3**: 1010 0111

**Q4**: FF

**Q5**: 25 (pronounced two-five, not twenty-five) and 64

**Q6**: 0011 1011 and 0001 0100

# Exercises

1. 0111 1011 [1]

2. 256 (0-255) [1]

3. 7B [1]

4. Hexadecimal values are much easier to remember than binary values. [2]

Hexadecimal numbers can be represented in only 2 character spaces, instead of 8.

5. 167 [1]

6. A7 [1]

7. 218 [1]

8 Allow any whole number that can be used in counting, e.g. one, two, three for a natural number.

An irrational number is any number that cannot be expressed as a fraction. For example, pi, or a non-repeating recurring value. [2]

9. All these values are part of the set ℝ of real numbers, since they can be expressed as a number with a decimal point. 22/7 would need to be converted to a decimal, however, as 3.142, for example. [1]

Chapter 14 - Bits, bytes and binary

# In-text questions

**Q1:**  The on/off symbol comprises a 0 and a 1 to represent a binary on or off..

**Q2:**  1000011 1100001 1110100

**Q3:**  5 characters x 16 bits (2 bytes) = 10 bytes.

**Q4:**  Parity bit = 0, **0**0010110

**Q5:**  01100110

**Q6:**  Answers will vary, but the barcode on the back of this book would be:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ISBN | **9** | **7** | **8** | **1** | **9** | **1** | **0** | **5** | **2** | **3** | **0** | **6** | **3** |
| *Weight* | *1* | *3* | *1* | *3* | *1* | *3* | *1* | *3* | *1* | *3* | *1* | *3* |  |
| Multiplication | 9 | 21 | 8 | 3 | 9 | 3 | 0 | 15 | 2 | 9 | 0 | 18 |  |
| Addition | *Add all the numbers* | | | | | | | | | | | | 97 |
| Remainder | *Find the remainder when divided by 10* | | | | | | | | | | | | 7 |
| Subtraction | *Subtract the result from 10* | | | | | | | | | | | | **3** |

# Exercises

1. (a) 011 0010 [1]

(b) 128 [1]

2. (a) ASCII uses fewer / 7 / 8 bits per character.

Unicode is a superset of ASCII. ASCII has only 128 (256) code points, whereas Unicode has over a million / uses 16/32 bits. [1]

(b) ASCII uses only 7 bits per character which will keep file sizes low.

ASCII cannot represent uncommon or foreign language characters. [2]

3. 256 MB = 0.25GB  
 1 TB = 1GB x 1024

1 TB = 0.25 GB x 4096

Award mark for a clear movement between MB – GB – TB making use of 1024;

Final answer: 4096;

Acceptable alternative (as many hard drive manufacturers do not use the 1024 principle):

1 TB = 1000 GB = 1,000,000 MB;

1,000,000/256 = 3906.25; (mark to be awarded for understanding the calculation needed)

Final answer: 3906.25; [2]

4. (a) 00011100 [2]

(b) Three times as much data must be transmitted. [1]

(c) 28 [1]

5. The first, or most significant bit, is set to either a 1 or 0; in order to make the total number of 1s in the byte an even number.

Accept any suitable example. [3]

Chapter 15 - Binary arithmetic and the representation of fractions

# In-text questions

**Q1:** Calculate 00100111 + 00011001.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **1** | **1** | **1** | **1** | **1** | **1** |  |  |
|  |  | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 39 |
| + |  | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 25 |
| = |  | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **64** |

**Q2:** 11001111

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **1** | **1** |  |  |  |  |  |  |
|  |  |  |  |  | 1 | 0 | 1 | 1 | 1 | 23 |
| + |  | 1 | 0 | 1 | 1 | 1 |  |  |  | 184 |
| = |  | **1** | **1** | **0** | **0** | **1** | **1** | **1** | **1** | **207** |

**Q3:** -32,768 to 32,767.

**Q4:** 10011.010

# Exercises

1. 1110 1101 [2]

2. 127 [1]

3. The number to subtract is converted into a negative number.  
This is then added to the first number.

Two marks for example:

49 = 00110001

1’s comp 11001110

add 1 1

-49 11001111

25 = 00011001  
 --------------

11101000

This is a negative number, find 2’s complement to translate to decimal:

1’s comp 00010111

1  
 ------------- 00011000 = 24 so result of original subtraction is -24  
 [2]

4. (a) 19 = 00010011 [2]

Flipped: 11101100 + 1 = 11101101

(b) -128 to 127 [1]

5. Convert and add 1: 01010100 + 1 = 01010101 = 85. Add the sign: -85. [2]

6. (a) (i) 01000100 (ii) 01110011 (iii) 01101110 [3]

(b) (i) 6.5 (ii) 3.125 [2]

(c) Largest: 4095.9375, Smallest: 0.0625 [2]

Chapter 16 - Bitmapped graphics

# In-text questions

**Q1**: 256 colours = 1 byte, 1024 x 1024 = 1,048,576 pixels. Answer = 1,048,576 bytes or 1MiB.

**Q2**: Position (coordinates) of the top left corner, width and height, line colour, line thickness, and fill colour.

# Exercises

1. (a) The number of pixels/dots; per cm/inch/unit of measurement; [2]

(b) The number of bits used to represent (the colour/greyscale value); [2]

(Reject number of (different) colours of a single pixel)

(c) 50;// 10\*10;\*4÷8;//100; ÷2;//100;\*0.5; [2]

(d) Does not deteriorate (A. Concept of deteriorating by implication) when enlarged/magnified // (usually) faster to transmit // (usually) faster to load // (usually) uses less memory/storage space // Easier to edit/manipulate objects in the image

(NE Easier to edit/manipulate) [2]

2. (Each pixel) can be one of 4/22 possible colours/values // Two bits are needed to represent the 4 possible bit patterns/colours/values // because there are 4/more than 2 colours in the image;

The second line of pixels (from the top) has been represented in a computer's memory as the bit pattern 1111 1100 0011 1111. A black pixel is coded as 11. [1]

(b)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

or

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

Mark as follows:  
 13th and 14th bits correct;  
 Other bits correct; [2]

(c) 8\*8 =64; \* 2 = 128; ÷ 8 = 16 bytes; //  
8\*8\*2÷ 8;  
16;  
128 bits [3]

(d) Coordinates of corner/corners // position of a corner // top left coordinates;  
Identifier;  
Length of side(s) // width // height // coordinates of an opposing corner;  
Line colour // outer colour;  
Line width;  
Fill colour // inner colour;  
Angle of rotation;

coordinates of midpoint/centre;  
radius/diameter

circle/oval

(Not enough) Position/coordinates  
(Not enough) Colour [3]

(e) (For geometric images) less storage space/memory likely to be needed;   
(For geometric images) will load faster from secondary storage;  
(For geometric images) will download faster;

Can be scaled/resized without distortion;  
Image can be (more easily) searched for particular objects;  
Can (more easily) manipulate individual objects in an image; [2]

Chapter 17 - Digital representation of sound

# In-text questions

**Q1:**  Graph 2 represents greater accuracy as there are more point values on the y axis at which a particular sample can be attributed. Graph 1 has some readings that have had to be placed near the original curve rather than on it owing to fewer points.

**Q2:**  Audio bit depth and the sampling rate both improve quality as they increase.

**Q3:**  10,000 samples per second x 8 bits = 80,000 bits per channel  
Stereo = 2 channels = 160,000 bits per second or 160kbits/s

**Q4:**  24KHz.

# Exercises

1. (a) 300; \* 2; // 600;

NOTE: award 1 mark for doubling an incorrectly calculated highest frequency [2]

(b) Regular samples are taken (of the analogue signal);  
Samples are quantised // the height of each sample is approximated to an integer value // height of samples measured // amplitude/volume measured;  
Each integer value is encoded as a binary value // measurements are coded in a fixed number of bits;  
output the binary numbers as digital signals/voltage levels; [3]

(c) Can (easily) synthesise musical notation from it;  
 Can be played on different instruments;  
 Can be (easily) transposed to a different key/pitch;  
 Produces (relatively) small files;  
 Easy to manipulate (the data);  
 Allows for easy interface with electronic musical instruments;  
 No data lost about a musical note; [1]

(d) Length/duration (of note) // Note-on and Note-off;  
Instrument;  
Velocity//Speed;  
Volume//Amplitude;  
Timbre;  
Pedal effects;  
Channel;  
Instructions about how to recreate a sound;  
Aftertouch;  
Pitch bend;  
Note envelope;

Reject: Note/key/pitch/frequency; [1]

2. (a) 16 (bit) // 2 bytes [1]

(b) 8,800,000 // 100 \* 2 \* 44,000  
44,000; [3]

(c) Because of Nyquist's theorem // Because we should sample at least double the highest frequency in the original sound; [2]  
Some people can hear higher frequencies than the average (so more than double has been chosen);  
There is no need to sample at a higher rate as humans won't notice any difference in quality above this level // sampling at a lower rate would mean that some people would notice the lower quality of the recording // sampling at a lower rate would mean that some meaningful changes in the analogue signal could be missed;  
higher rate would require more, unnecessary, storage space; [2]

(d) Compression has been used;

Explanation of a particular compression method that could have been used on the recording e.g. lower sampling frequency used // lower sampling resolution used;

[1]

3. (a) 6/100; //600; [2]

(b) 8 (bits); A. 1 byte; [1]

(c) Sample at a frequency (at least) twice the rate; of the highest frequency (that can be present in the original signal); [2]

Chapter 18 - Data compression and encryption algorithms

# In-text questions

**Q1**: Website image – lossy, a zipped file – lossless and a PDF – lossless.

**Q2**: OZQNZX HFJXFW

**Q3**: BADGER

**Q4**: 2

# Exercises

1. Lossy compression permanently removes unnecessary data giving only a close approximation to the original file after decompression. Perceptually, this may look the same, but a byte-by-byte comparison will show differences.

Lossless compression uses an algorithm to create a compressed version of a file with all of the instructions required to replicate it again as an exact copy. [2]

2. (a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |

[3]

(b) Although compressed compared to the original, lossless methods result in larger files than with lossy compression. [1]

(c) Transmission of data is faster (web pages will load more quickly) and uses less data allowance with smaller image files. [2]

3. (a) Since lossy methods do not retain 100% of the original data, a text file would lose characters and become difficult, if not impossible to read. [1]

(b) Dictionary-based compression. [1]

4. (a) An encryption key that is only used once. [1]

(b)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **e** | **£** | **XOR F** | **x** | **]** | **XOR**  **%** | **a** | **F** | **XOR**  **‘** | **m** | **L** | **XOR**  **!** |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

Ans: **F%'!** [4]

(c) The Vernam cipher uses a **truly random key** which is **equal to or longer in characters than the plaintext**. This creates a truly uniform distribution of ciphertext characters and makes it impossible to find any pattern. Other ciphers use mathematics to generate a key which, in theory, can be broken by reversing the process. [2]

Chapter 19 – Hardware and software

# In-text questions

**Q1**: Hardware is the term used to describe the physical parts of a computer and its input, output and storage devices.

e.g. CPU, RAM, ROM, keyboard, monitor, hard disk

Software comprises all the programs that are written to make computers function.

e.g. operating system, word processor, Python program

**Q2**: Depends on the programming language. In Python, for example, to use a library, e.g. “random”, write import random at the start of the program.

# Exercises

1. (a)(i) System software is the software needed to run the computer’s hardware and application programs. [1]

(a)(ii) Application software is any program designed to perform a specific function directly for the user. [1]

(b)(i) Operating system, utility programs, libraries, programming language translators. [1]

(b)(ii) Word processor, spreadsheet etc. (NOT Word, Excel, or other specific products) [1]

2.

|  |  |
| --- | --- |
| **Software** | **Category** |
| Firewall software installed on the web server | **Utility software** |
| Store’s own online ordering system designed for their products and systems | **Bespoke software** |
| Graphics software to crop product images suitable for uploading to the site | **General purpose application software** |
| Online payment verification software | **Special-purpose software** |

[4]

3. It will be less expensive

It will be available immediately

It will be thoroughly tested and more likely to be bug-free

Guidance and support is likely to be available in textbooks and online forums [2]

Chapter 20 – Role of an operating system

# In-text questions

**Q1**: It means that the processor can swap rapidly from one task to another, giving the impression that several tasks are being done simultaneously. It is not the same as multiprocessing, which means having more than one processor.

**Q2**: Response time does slow down: Installing more RAM will improve the performance of the computer, because the operating system will spend less time swapping programs and data between RAM and hard disk (virtual storage).

**Q3**: “Round Robin” means it gives each process an equal time before moving on to the next, unless an interrupt occurs, for example a process is waiting for user input or needs to print.

Another algorithm is “shortest job first” where short jobs take priority, but this requires knowledge of how long a job will take.

**Q4:** File access time slows down because it has to look for different blocks in different areas of the disk so the read-write head may have to physically move to read each block.

Unwanted files need to be deleted and a defrag utility run.

**Q5:** Processor is waiting for user input

User cancels program by pressing Esc

Run-time error such as division by 0 or non-existent file requested

# Exercises

1. An operating system is designed to hide the complexities of the hardware from the user and to manage the hardware and other resources. [2]
2. Processor management // Allocation of processors // Allocation of processor time // (process) scheduling // thread management; A processing management

Allocation/management of RAM / memory // allocation of buffers;

Allocation/management of / control of I/O devices/peripherals // I/O management // device driver management;

File / backing store / secondary store management / access / organisation;

Power / battery management;

Interrupt handling;

Provision of Application Program Interface / API; An interface between hardware and applications

Provision / management of (windows in) user interface;

Management of system security;

Answers by example, only one example of each type

A description of a type of software management but not just “software management”. e.g. loading of programs, software installation, registering DLLs. [3]

Chapter 21 – Programming language classification

# In-text questions

**Q1**: 16

|  |  |  |  |
| --- | --- | --- | --- |
| **Accumulator** | **Location 8** | **Location 9** | **Location 10** |
| 25 | 25 | 23 | 25 |
| 23 | 23 | 23 | 25 |
| 25 | 23 | 25 | 25 |

**Q2**:

**Q3**: LDA 8

ADD 9

STO 10

# Exercises

1. (a)( i) Indicates the basic machine operation/function/command;

Executable binary code; **Or**, “instruction” – with a valid example [1]

(a)(ii) Represents a single item of (binary) data / a single value;

Represents a memory address / storage location;

The value that the instruction operates on;

**Or**, A parameter for the operation [1]

(b) Easier to understand;

Takes less time to code (as using mnemonic opcodes and hex operands);

Fewer mistakes made in coding;

Ability to add comments to code;

Use of symbolic names for operands // easier to remember opcodes;

Use of labels;

Easier to maintain/debug; [2]

Chapter 22 – Program language translators

# In-text questions

**Q1**: Retention of source code ensures control over it is kept with company or individual; software cannot be so easily reverse engineered; code cannot be modified.

# Exercises

1. Assembly code is an example of a low-level language. Each assembly code statement, generally speaking, translates into one machine code instruction – this process is performed by an **assembler**. The statements are written using mnemonic instruction codes for Add, Load, Store etc.

Pascal, Python etc. are examples of high level languages.

Each instruction translates into several machine code instructions. A **compiler** is used to translate a high level language into machine code.

Advantages of assembly code: Can occupy less space, may run faster, programmer has control over individual bits. Appropriate in some embedded control systems

Advantages of high level language – faster to learn, write, easier to understand and maintain.

Is not machine-specific and can be run on different hardware.

The code can be compiled and distributed without the source code

Appropriate for the vast majority of application programs and even for writing a compiler program. [10]

2. A big advantage of bytecode is that you can achieve **platform independence**; the bytecode can be interpreted on different types of machine architecture.

A second advantage of using, for example, Java bytecode is that it acts as an extra security layer between your computer and the program. You can download an untrusted program and you then execute the Java bytecode interpreter rather than the program itself, which guards against any malicious programs.

It is also possible to compile from Python into Java bytecode (using the Jython compiler) and then use the Java interpreter to interpret and execute it. [5]

3. (a) Load B

Add #5

Store A [3]

(b) (i) Assembler [1]

(b) (ii) Compiler or interpreter [1]

Chapter 23 – Logic gates

# In-text questions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input A** | **Input B** | **Input C** | **D = B OR C** | **E = A AND D** | **Q = NOT E** |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |

**Q1**:

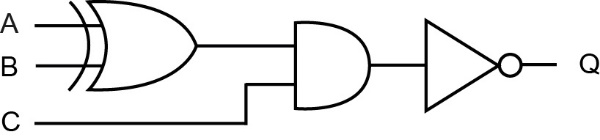
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input A** | **Input B** | **NOT B** | **C = A AND (NOT B)** | **D =(NOT A) AND B** | **P = C + D** | **Q = A XOR B** |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |

**Q2**:

**Q3**: Comparing the AND and NAND gates, all the 0s become 1s and all the 1s become 0s.

Similarly, comparing the OR and NOR gates, all the 0s become 1s and all the 1s become 0s.

**Q4:**

****

# Exercises

1. (a) NAND gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output Q** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

[1]

NOR gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output Q** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

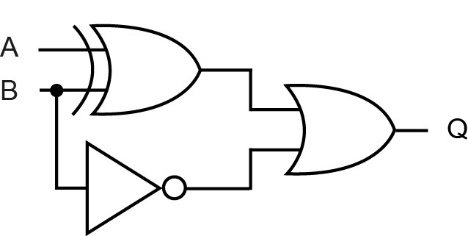
[1]

XOR gate

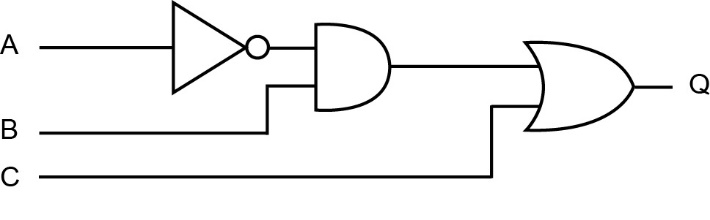
|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

[1]

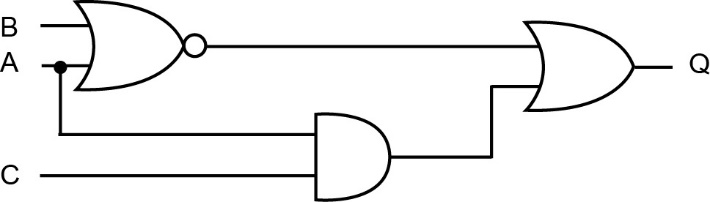
(b)(i) [3]



(b)(ii) [3]



(b)(iii) [3]



2. (a) [3]

(b) F = 1, G = 1, H = 0, K = 0, Q = 1 [5]

Chapter 24 – Boolean algebra

# In-text questions

**Q1**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** |  |  | **A + B** |  |  |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**Q2:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** |  |  | **A ⋅ B** |  |  |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**NAND symbolQ3:** (i)

A

B

Q

NOR symbol (ii)

A

B

Q

**Q4:**   
 = [since according to de Morgan’s theorem ]  
 = [since ]  
 =

Therefore .

**Q5:**

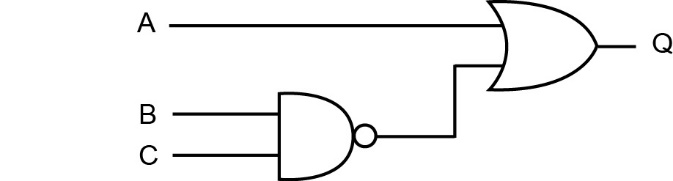
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A + B** | **A + C** | **(A+B)(A+C)** | **A+B C** | **B C** | **A** |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Q6:** A (A + B) = A (1 + B) (since AA = A from Law 2)

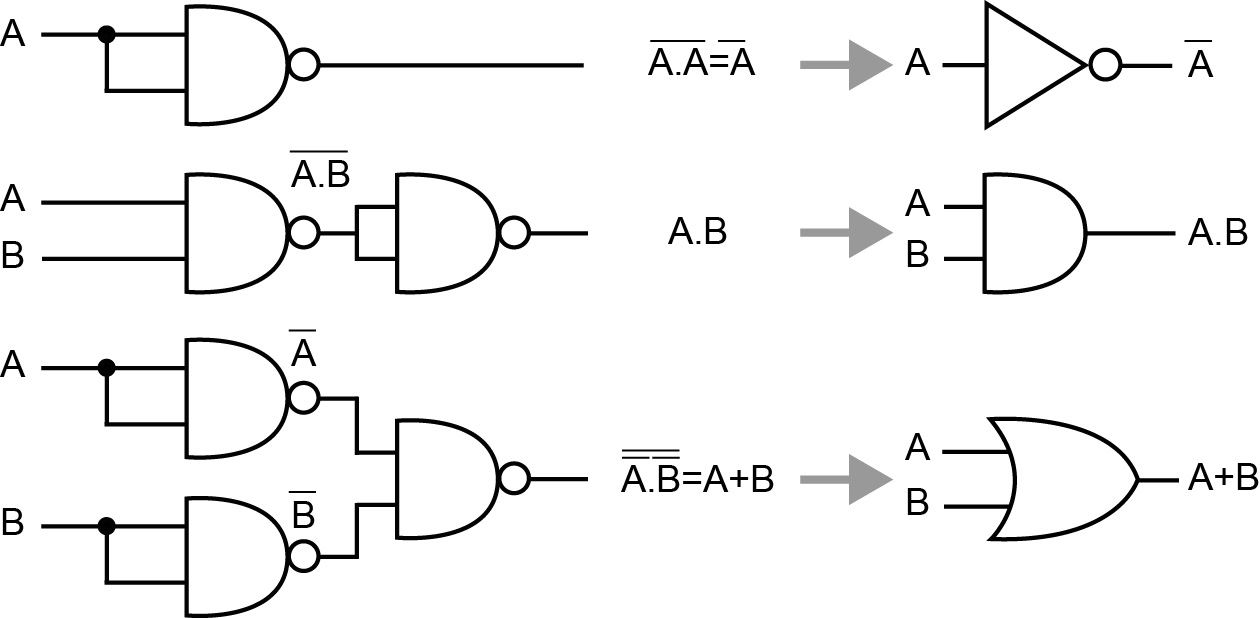
= A (since 1 + B = 1 from Law 6)

= 1

**Q7:**



**Q8:**



# Exercises

1. (a) X Y;

**A** alternative notations : X XOR Y

X EOR Y

X AND NOT Y OR NOT X AND Y [1]

(b)

**A** alternative notations : X AND NOT Y; [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | **Outputs** | |
| **X** | **Y** | **C** | **S** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

(c)(i)

[2]

(c)(ii) Addition // adder; [1]

**A sum;**

(d) [Fully expanding brackets – **1 mark**]

[Recognising OR –1 mark]

[Taking X outside brackets – **1 mark**]

OR

Final answer, [**1 mark**]

**Alternative Answer : (Distributive)**

[Use of distributive law – **1 mark**]

[Recognising - **1 mark**]

[**1 mark**]

[Final Answer, **1 mark**]

**Alternative Answer : (De Morgan’s)**

[Use of De Morgan's – **1 mark**]

[Two further applications of De Morgan's]

[Taking X outside brackets – **1 mark**]

[Recognising = 1 – **1 mark**]

[Final answer, **1 mark**]

[3]

2. (a)

|  |  |  |
| --- | --- | --- |
| **AND Gate** | | |
| **Input X** | **Input Y** | **Output Q** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **XOR Gate** | | |
| **Input X** | **Input Y** | **Output Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1 mark for each of the output columns [2]

(b)(i)

[Brackets are not necessary]

1 mark for use of correct operands (L,R,U);

1 mark for use of XOR with L,R;

1 mark for NOT U anded with other part;

Alternative:

1 mark for use of correct operands (L,R,U);

1 mark for alternative XOR expression;

1 mark for AND NOT U;

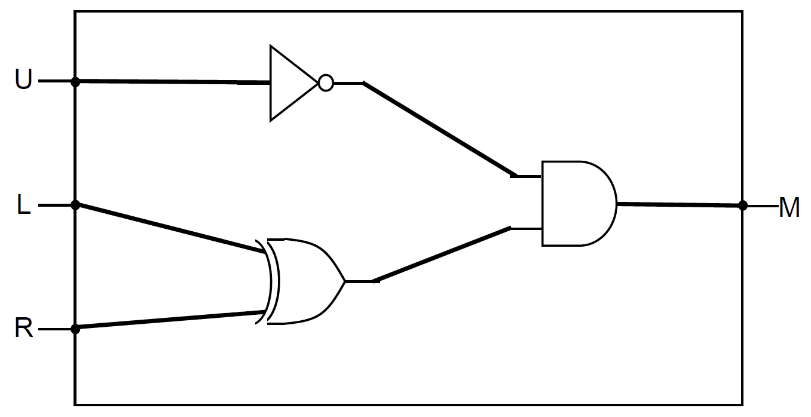
Alternative:

1 mark for use of correct operands (L,R,U);

1 mark for alternative XOR expression;

1 mark for AND NOT U; [3]

(b)(ii)



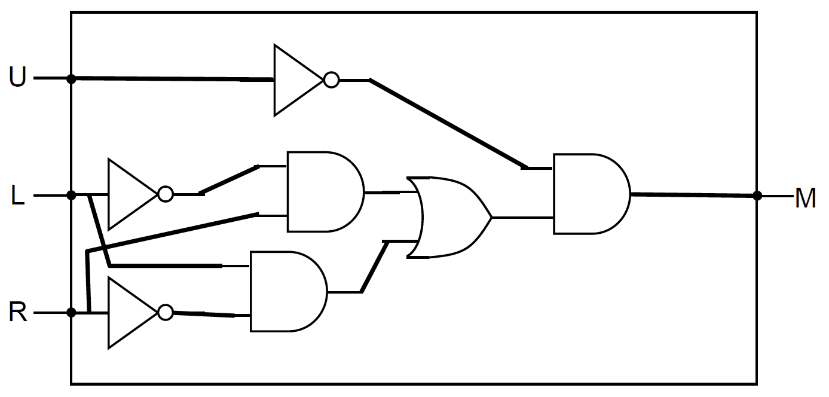
L, R connected to XOR gate;

U connected to NOT gate;

Output of a two input AND gate connected to M;

**MAX 2** if circuit does not reflect the correct logic

**Alternative :**

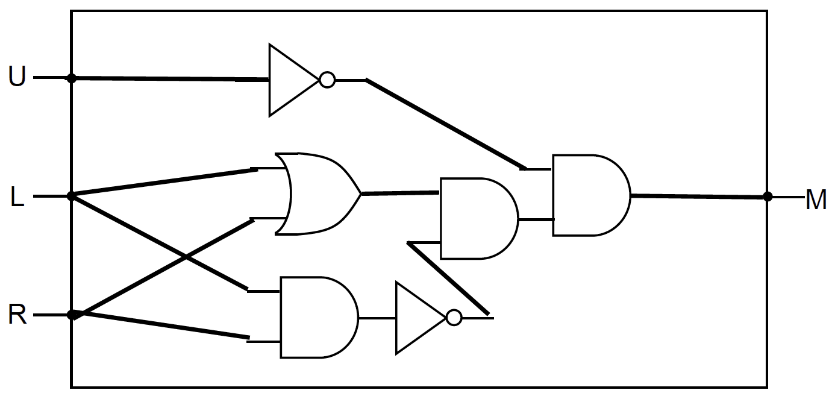


U connected to NOT gate;

Correct gates used for L and R before last AND gate;

Output of a two input AND gate connected to M;

**Alternative :**



Marked as above alternative. [3]

(c) **Solution1:**

[Application of De Morgan's Law –**1 mark]**

[allow simplification of double nots at same time] [Simplification of A.A to A – 1 mark]

[Correct solution – **1 mark**]

**Solution 2:**

[Application of De Morgan's Law – 1 mark]

[allow simplification of double nots at same time]

[Simplification of NOT A OR NOT A to NOT A – **1 mark**]

[De Morgan's again to correct solution – **1 mark**]

No working marks for truth table solution (asked to use De Morgan's in question)

[3]

3. (a) AND;

NOR;

XOR; accept EXOR, EOR, NEQ, exclusive OR [3]

(b) (i) B [1]

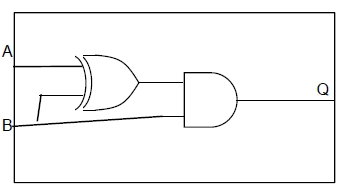
(ii) B [1]

(iii) = 0

or = 0

(give 1 mark if not simplified to 0) [2]

(c)



[2]

Chapter 25 – Internal computer hardware

# In-text questions

**Q1**: The keyboard controller and the VDU controller. The keyboard cannot accept input and the VDU cannot output data.

(i) mouse controller – from the data bus to the mouse

(ii) bi-directional, since a touchscreen is both input and output.

**Q2:** No! The size of a stick of 32GB RAM is about 6 inches by 0.5 inches. To reach 16.8 million terabytes, if you put the 32GB sticks end to end, would require a motherboard of 4,242 miles.

<http://www.howtogeek.com/175443/what-is-the-maximum-amount-of-ram-you-could-theoretically-put-in-a-64-bit-computer/>

# Exercises

1. (a) A set of/group of/parallel wires/lines;

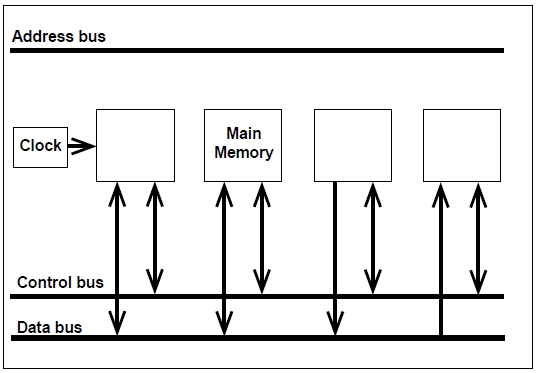
that are used to connect together components (inside the computer) // connect different parts of the CPU;

in order to pass signals between them; [2]

(b) The data bus can be used to transfer data and instructions between the main memory and the processor. The control bus carries control signals.

An example of a control signal is:

Clock/timing signal; reset signal; I/O write; I/O read; memory read; memory write; transfer ACK [5]

[5]

graphics controller

keyboard controller

Processor

2. (a) Machine code; [1]

(b) 0 and 255 (FF); [2]

(c) 8 lines; [1]

Chapter 26 – The processor

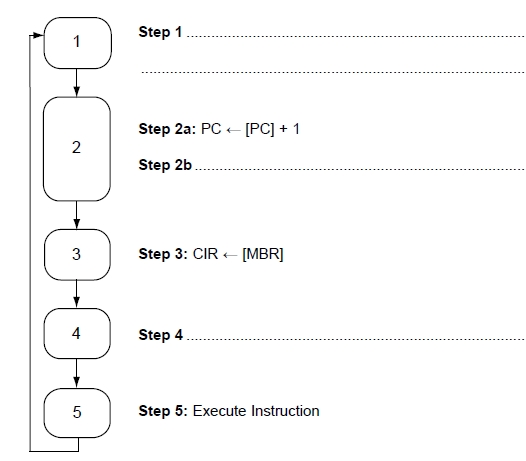
# In-text questions

**Q1**: See exercise 1(b) below.

**Q2**: If the instruction is a JUMP or BRANCH instruction, the address to branch to, will be copied to the PC.

# Exercises

1. (a) Register 1/2 ....Program Counter, Current Instruction register, Memory address register, memory buffer/data register, status register [2]

 (b)

Step 1: Copy address from PC to MAR

Step 2b: Copy instruction from address held in MAR to MBR

Step 4: Decode instruction held in CIR [3]

2. (a)

|  |  |
| --- | --- |
| **Component number** | **Component name** |
| 1 | Memory address register |
| 2 | Data bus |
| 3 | Control bus |

[3]

(b) To fetch / decode / execute instructions;

To synchronise operation of processor;

To marshal/control operation of fetch-execute cycle;

To send control signals/commands to other components of fetch-execute cycle;

To control the transfer of data between registers/MBR; [1]

(c) Arithmetic logic unit; [1]

(d) A (very fast) memory location within the processor; [1]

(e) Arithmetic results – Overflow/underflow/positive/negative/zero/carry;

Interrupts (enabled/disabled);

Parity;

BCD arithmetic enabled/disabled;

Supervisor mode;

Halt; [1]

Chapter 27 – The processor instruction set

# In-text questions

**Q1**: Store the contents of the accumulator in memory location 33.

**Q2**: Load the contents of location 12 into the accumulator.

**Q3**: 15.

**Q4**: 16.

**Q5**: TOTAL = 12 + 2

**Q6**:

|  |  |  |
| --- | --- | --- |
| **Machine code** | **Assembly code** | **Meaning** |
| 0100 1010 | LDA 10 | Load the contents of location 10 into the accumulator |
| 0010 1011 | ADD 11 | Add the contents of location 11 to the accumulator |
| 0111 1110 | STO 14 | Store the result from the accumulator in location 14 |

# Exercises

1. (a) The set of instructions that the CPU can execute; [1]

(b) 26, i.e. 64; [1]

(c) 210- 1 = 1023; [1]

(d) The number of opcodes would be increased to 256;

The largest operand would be reduced to 255; [2]

(e) More space available for the opcode, giving a larger instruction set,   
and a greater range/precision for the operand [2]

2. LOAD Y

ADD #5

STORE X [3]

3. (a) Machine code; [1]

(b) Memory location/address; [1]

(c) Opcode; operand; [2]

(d) One-to-one. One assembly code instruction translates into one machine code instruction. [1]

Chapter 28 – Assembly language

# In-text questions

**Q1:** (i)CMP R1, #100

(ii) BNE .loop1

**Q2:** 264 = (1)00001000. The overflow bit in the status register will be set to 1

**Q3:** AND R3, R4, #10111111

**Q4:** In the status register.

**Q5:** Instruction description: “Logical shift left, the value in R2 by 3 places, store in R3”

R2 will hold its original value, R3 will hold 10011000

# Exercises

1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIRST** | **SECOND** | **THIRD** | **R0** | **M** |
| 17 | 25 | 19 | 17 |  |
|  |  |  | 25 | 25 |

It puts the maximum of the three integers in M. [3]

2. (a) 26 = 64; [1]

(b) In immediate addressing the operand field contains the data rather than the address of the data. E.g. LDR R0, #54 loads the number 54 into R0

In direct addressing the operand field contains the address of the data. E.g. if memory location 54 contains 200, the instruction LDR R0, 54 loads the number 200 into R0; [4]

3. (a)(i) LOAD = Opcode, 4 = operand [1]

(a)(ii) A memory location in the processor [1]

(b) LOAD 12

ADD 13

STORE 14

(can swap operands 12 and 13) [3]

4. Bit 0 is the rightmost bit. B indicates binary number # indicates immediate addressing.

AND A, #01101001b ;AND contents of accumulator A and given binary

;number, storing result in A

CMP A, #01101001b ;compare contents of A with given binary number

BEQ LABEL1 ;branch if equal to LABEL1 [3]

Chapter 29 – Input-output devices

# In-text questions

**Q1:** f5

**Q2:** Mobile phones tend to use CMOS sensors as they consume far less battery power.

# Exercises

1. (a) For Photodiode System:

Light/laser/LED/Infra-red light shone at bar code; NE beam

(Moving) mirror/prism moves light beam across bar code//user moves reader across bar code; NE beam

Light reflected back;

Black/white bands reflect different amounts of light // black reflects less light // white reflects more light;

Light sensor/photo sensor/photo diode/CCD measures amount of reflected light;

Light reflected converted into an electrical signal; convert reflection to (binary) numbers/characters

(Electrical form of) reflection analysed to determine value encoded in bar code;

Data transmitted as binary codes to till/computer;

These values are often sent as ASCII codes;

For Camera/CCD System:

Camera/CCD measures (ambient) light reflected from bar code;

Camera/CCD converts light into an electrical signal;

Light reflected back;

Black areas reflect less light than white;

Raw image data transmitted to computer;

Image analysis software analyses image to determine value encoded in bar code;

MAX 4 [4]

(b) Validate data entry//check bar code is valid/reasonable;

Verify if bar code has been “input” accurately/correctly //check bar

code not damaged/altered;

**R** validate the item. [1]

(c) Keyboard/Keypad/Touch screen/Concept Keyboard/Electronic Scales

**NE** scales [1]

2. 1. reader sends radio frequency energy/wave;

2. to the antenna of the RFID tag in the book;

3. The RFID tag is energised by the reader/this energy;

4. the transponder (in the tag) sends the data signal;

5. the reader near the exit receives the data signal; [2]

3. RFID reader/scanner (at passport control) transmits/sends signal;

Signal which activates/energises/induces current RFID transponder/tag;

RFID transponder/tag transmits/sends data by radio (wave);

Electrical/physical contact between tag and reader not required//tag must be near to reader;

Passport may need to be unlocked using Machine Readable [2]

Chapter 30 – Storage devices

# In-text questions

**Q1:** Research task

# Exercises

1. (a) Data bits are recorded by “burning” a pit in the disk’s surface; recordable disks use a reflective layer with a transparent dye coating that becomes less reflective when a spot laser “burns” a spot in the track.

A change from a pit to a land (or vice versa) indicates a 1 (or 0); anything else is a 0 (or 1);

A high powered laser is used to “burn” the pits; a lower powered laser is used to read the disk; [3]

(b) No hardware exists to read CD-R disks;

The CD-R medium has become corrupted // CD-R is scratched/damaged/degraded;

Support for file format no longer available // no software capable of reading format data stored in CD-R; [2]

2. (a) Magnetic (medium);

Binary digits/bits/0s and 1s/data represented by magnetising spots on disk // changing magnetic properties of disk;

Disk made up of platter(s);

Disk divided into tracks and sectors; A. either tracks or sectors alone

Tracks are concentric circles // organised into cylinders;

Drive head can move in/out // moves to track/cylinder // moves radially;

Disk continuously spinning (while in operation);

Disk spins at high speed // feasible example of speed;

Data read/written as correct sector passes under read/write head; A. drive head

Data transferred in sectors/blocks;

Medium and drive/device in sealed enclosure;

Hard disk drive is a random access device;

**A**. Head parked / not over disk when not in use // head must not touch surface when in use;

**A**. Use of cache/buffer to speed up data transfer;

MAX 3 if candidate talks about lasers / making holes / pins / engraving [4]

(b) 512 MiB x 2 = 1024 MiB = 1GiB

1GiB x 1024 = 1 TiB

2 x 1024 = 2048

Award mark for a clear movement between MiB – GiB - TiB making use of 1024; [1]

Final answer: 2048;[1]

**Alternative**

240 / 229; = 211;

(c) More platters (which are packed closer);

Greater density of data on each platter;

More tracks on a platter // more cylinders;

Change to perpendicular magnetic domains;

Ability to write smaller magnetic domains/parts // smaller read/write heads;

Use of different alloy materials for the platters; [1]

(d) Faster access speed // faster booting of operating system // faster data transfer/read/write speeds;

Silent operation;

Are lighter;

Less heat generated;

Less power required // longer battery life;

Less susceptible to damage from physical shocks // more robust (due to no moving parts);

**Not Enough** quicker (without explanation)

**NE** better performance (without explanation)

**Accept** – quicker as no need to wait for read/write head to move//sector to be underneath read/write head; [2]

Chapter 31 – Communication methods

# In-text questions

**Q1:** Speed test

**Q2:** P: parity bit = 0

Q: parity bit = 1

# Exercises

1. (a) Serial send one bit at a time / after each other

whereas parallel sends multiple bits simultaneously/at same time;

**Accept** "data" for "bits" in the context of parallel transmission

Serial uses a single wire/cable/path/line whereas parallel uses several/multiple

wires/cables/paths/lines;

**Reject** answers that refer to multiple channels achieved by sharing bandwidth

Both sides of point must be made. [2]

(b) Parity Bit: 1;

Start bit, Stop Bit : Can be either 0 or 1, but must both be different to get mark; [2]

(c) Receiver and transmitter (clocks) do not need to be/are not (exactly) synchronised // transmission of data without use of external clock signal // receiver and transmitter clock only synchronised at start of/for length of transmission // start bit used to synchronise clocks of sender and receiver // data sent as soon as available rather than waiting for clock pulse/synchronisation symbol; [1]

2. (a) Bit rate – the speed at which data is transmitted serially, measured in bits per second (bps).

Baud rate (also called the symbol rate) – the rate at which the signal changes.

There is a direct relationship between bit rate and baud rate.

bit rate = baud rate x number of bits per signal [3]

(b) Latency is the time delay between the moment the first byte of a communication starts and when it is received at the destination. [2]

3. (a) A set of rules relating to communication between devices [1]

(b) Protocols are necessary to ensure that equipment from different suppliers can communicate [1]

(c) Physical connections, cabling, mode of transmission, speed, data format, error correction. [5]

Chapter 32 – Network topology

# In-text questions

**Q1:** You can read the article at http://www.howtogeek.com/196998/your-devices-broadcast-unique-numbers-and-theyre-being-used-to-track-you/

“Garbage cans were placed all over the city, and Wi-Fi monitoring hardware was installed in them. Then, the garbage cans were networked together. When you walked by one of these garbage cans, your device would send probe requests with its MAC address and the garbage can’s sniffer would make a note of the MAC address and its location. When you walked by another garbage can, it would note your device’s MAC address and location again. This information could be combined to form a picture of your movements throughout the day. Advertisers would know the areas you visited and could try to target ads specifically to you. With enough Wi-Fi sensors joined together, it would be possible to track your smartphone’s complete movements over an entire day.”

A store could place Wi-Fi sniffers throughout their store and log MAC addresses. Perhaps you spent some time in the electronics section before leaving for another section of the store — the store could display ads for electronics to you.

Security services and police can track the movements of individuals they are interested in.

**Q2:** MAC addresses are assigned at the factory, but it is possible to change a MAC address in software.

Networks can use MAC address filtering to allow only devices with specific MAC addresses to connect to a network – someone might want to hack into a network by changing their MAC address to one that is allowed. Students may research other reasons, most of them fairly questionable!

# Exercises

Server

Terminal

Hub or Switch

**Star Network**

**Bus Network**

Server

Terminal

1. (a)

[4]

(b) Advantages of star topology:

If one cable fails, only one station is affected, so it is simple to isolate faults

Consistent performance even when the network is being heavily used

Performance is better than a bus network, with speeds of up to 100mbps

No problems with ‘collisions’ of data since each station has its own cable to the server

The system is more secure as messages are sent directly to the central computer and cannot be intercepted by other stations

Easy to add new stations without disrupting the network

Different stations can transmit at different speeds

Disadvantages of star topology:

May be costly to install because of the length of cable required

If the central device goes down, the whole network will go down. [4]

(c) a network physically wired in star topology can behave logically as a bus network by using a bus protocol and appropriate physical switching. [2]

2. (a) A MAC address is a number assigned and hard-coded into a network interface card (NIC) by the manufacturer which uniquely identifies the device. [1]

(b) The student is largely correct. For example, when you walk around with a smartphone, it scans for nearby Wi-Fi networks and broadcasts its MAC address. However in some network interfaces, the MAC address is set in software and can be changed, though this would be unusual. [3]

Chapter 33 – Client-server and peer-to-peer

# In-text questions

**Q1:** Literary work, dramatic work, musical work, artistic work, films, sounds recordings, broadcasts, typographical arrangement of published additions, software.

Copyright lasts for 70 years after the death of the creator.

**Exercises**

1. In a client-server network, one or more computers known as **clients** are connected to a powerful central computer known as the **server**. Each client may hold some of its own files and resources such as software, and can also access resources held by the server.

In a peer-to-peer network, there is no central server. Individual computers are connected to each other, either locally or over a wide area network so that they can share files.

Client-server network could be used in an organisation such as a school, hospital, or small, medium or large business.

Peer-to-peer network in someone’s home or small office. (Also used for streaming of movies, e.g. video on demand.) [6]

2. Some organisations are set up to illegally distribute copyright music and film without permission. [1]

Consequences are that the artists, film makers, and others involved in the making do not get royalties for their work. Film makers and music companies have less money to produce new works. [5]

Chapter 34 – Wireless networking, CSMA and SSID

# In-text questions

**Q1:** Benefits: can easily find a route to a particular location. Can find out when the next bus arrives, etc.

**Exercises**

1. (a) A computer device with a wireless network interface controller, and a router to connect to the Internet. [2]

(b) WPA2 provides strong encryption of data transmissions.

SSID requires a user of a local network to type in a password (security key) before they can use the network. [4]

(c) Prior to transmitting, a node first listens for signals on the wireless network to determine whether another node is transmitting. If a signal is detected, it waits for a random period of time for the node to stop transmitting and then listens again. [4]

(d) Having determined that no other node is transmitting, a **Request to Send/Clear to Send (RTS/CTS)** may optionally be used before transmitting. This counteracts the problem of “hidden nodes”, i.e. a node that can be heard by the WAP but not by the node trying to transmit.

A “request to send” signal is transmitted first, and the data itself is not transmitted until a “Clear to send” signal is received back. [2]

2. Wi-Fi location based technology: there are several methods.

e.g. for phones without GPS features, the service provider gets the location based on the radio signal delay of the closest cell-phone towers to determine the cell phone’s location. In the UK, networks do not use trilateration; LBS services use a single base station, with a "radius" of inaccuracy, to determine a phone's location.

By knowing where their customers are, retailers can:

help them get to their shop

tell them about special offers at the shop

advertise particular products

They can use apps which users download, to target users with particular products, deliver tailored content and create individualized in-store experiences for customers.

Stores can gather statistics on how many people visit the store, how long people stay, which particular areas are most visited, etc.

Dangers: there is a privacy issue – you could be tracked by a stalker, or a burglar would know you were not at home, etc. [8]

Chapter 35 – Communication and privacy

# In-text questions

**Q1:** The village chief may be the person to consult. Or, would you go over his/her head and ask the local government official? You would have to find out when you were in the area concerned.

You should ask whether there were any objections to the plan, so that these should be addressed, and local people reassured.

Benefits – It could mean that people bring water, power or roads to the village would be able to plan routes.

Risks – it could be seen as an invasion of privacy. Terrorist and extremist groups can be quite computer savvy and may research where villages, schools and houses are before planning an attack.

Safeguards: Possibly individuals should not be recognisable.

These are just a few ideas for discussion – you and the students may come up with others!

**Exercises**

1. Arguments for: Whistle blowers perform an essential service in reporting illegal activities of their employers.

Arguments against: Snowden had signed the Official Secrets Act – should we assume that the NSA was acting in the best interests of US citizens and protecting them from terrorist attacks? Maybe the end (preventing an attack) justified the means, (collecting private data) [4]

2. Strong encryption is necessary for banks etc. to ensure their data is not hacked.

On the other hand it is necessary for security forces to be able to decrypt terrorist or criminal activity on the Internet which may lead to loss of life.

Maybe some organisations should be given permission for strong encryption but not individuals. [4]

3. Modern warfare is quite possibly going to be waged via the Internet rather than by dropping bombs – imagine the effects of bringing down a country’s banking systems, food distribution networks, etc.

It is best to be prepared with measures of your own, just as it is necessary for countries to be able to defend themselves against armed attack.

Being prepared can have a strong deterrent effect against other countries planning a cyber-attack. [4]

Chapter 36 – The challenges of the digital age

# In-text questions

**Q1:** Discussion and research question.

**Exercises**

1. Yes, they should. They make an enormous amount of money from advertising and they should invest in measures to protect users from trolls and cyber-bullying.

Should not allow content which abuses individuals, constitutes cyber-bullying, is libellous or malicious.

Yes, it is possible to develop software to detect many forms of abusive content. Facebook has recently developed software to do this. [5]

2. Bus, taxi and truck drivers may be replaced by driverless cars.

Job centres may have fewer staff as people apply for jobs online

Financial advisers may not be needed by as many people as they use sites like comparethemarket.com, etc.

(Many other answers in the text)

Social effects: Middle-income jobs will disappear, but may be replaced by more people starting businesses which use computer technology, or working in new computer-related jobs which replace the old jobs.

Maybe more people will work from home, will have more flexible hours, etc. (Accept all reasonable answers!) [5]

3. No marks are given for this question, which is intended to get students to do some research and practise their essay-writing skills. Some suggestions are:

The software engineers presumably engineered the software so that it would pass the emissions tests, egged on by the engineers who designed the engines. BUT the chief exec of the company is ultimately responsible. If there is a culture of “we must make a good profit whatever it takes” then employees will do what it takes to please the bosses.

The motivation of the perpetrators was probably to “get the job done”, maybe get a bonus, maybe they did not believe the emissions tests were important as a measure of environmental pollution…

The software engineers certainly did great harm to the company, which has had to recall millions of cars, and great inconvenience to the millions of owners who have had their cars recalled. The stock price of VW dropped, causing pain and grief to shareholders (or their pension funds which quite likely had shares in VW). Whether it has done major harm to the environment is unclear.

4. The algorithms are probably not fair to all – the data is often ambiguous and may not take into consideration all the facts. For example to take into account parents’ medical history would be extremely tricky. If three close relatives died at a relatively young age from cancer, does that have any bearing on your own life expectancy? Probably not. If your occupation is teacher, are you more or less likely to have an accident than a gardener? It is all worked out statistically but there are so many factors the resulting algorithms are probably often biased or inaccurate.

Taking into account your address seems questionable – is it culturally biased? Possibly.

Chapter 37 – Queues

# In-text questions

**Q1:**

|  |  |  |
| --- | --- | --- |
| **Queue operation** | **Queue contents** | **Return value** |
| q.isEmpty() | [] | True |
| q.enqueue(‘Blue’) | [‘Blue’] | - |
| q.enqueue(‘Red’) | [‘Blue’,‘Red’] | - |
| q.enqueue(‘Green’) | [‘Blue’,‘Red’,‘Green’] | - |
| q.isFull() | [‘Blue’,‘Red’,‘Green’] | False |
| q.isEmpty() | [‘Blue’,‘Red’,‘Green’] | False |
| q.deQueue() | [‘Red’,‘Green’] | ‘Blue’ |
| q.enqueue(‘Yellow’) | [‘Red’,‘Green’,‘Yellow’] | - |

Note that items being added to the front of the queue are appended to the end of the list or array in this implementation.

**Q2:**

rear = 5

front = 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eli | Jason | Milly | Bob | Adam | Jack |

Three names are in the queue and there are no spaces left for any more items to join.

**Q3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

rear = -1

front = 0

There are different options for initialising pointers – these are initialised as in the algorithm SUB initialise. When the first item is added, both pointers point to it, as it is both at the front and rear of the queue.

rear = 0

front = 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Greg | Ben | Charlie | Davina | Enid | Fred |

**Q4:** It is an abstraction because the queue is *logically* but not *actually* circular, it is a linear array in memory – and even the concept of a linear array holding values is an abstraction, as it is just made up of circuitry inside the computer. Furthermore, the items are not actually removed from the list; the pointers tell us where the logical start and end of the queue are.

**Q5:** An item would join the queue at the front if it were a high priority item, and there were no other high priority items already in the queue. It would join at the rear if was categorised as a “lowest priority” item.

Exercises

1. (a) It may be implemented as a circular queue if a static data structure such as an array is being used, so that as items leave the front of the queue new items can take their place as the rear pointer “wraps round” to the front. [2]

(b) A dynamic data structure can expand as needed, memory locations being taken from the heap as required.

This type of structure is useful for implementing a queue in which the maximum number of items is not known.

The heap is a portion of memory which can be allocated or deallocated as required. [4]

(c) (i) [3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Job1 | Job2 | Job3 | Job4 | Job5 |

rear=4

front=0

(ii) [2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Job6 | rear=0  front = 2 | Job3 | Job4 | Job5 |

2. (a) Values/cards need to be taken out of the data structure from the opposite end that they are put in // cards removed from top/front and added at end/bottom/rear; Values/cards need to be removed in the same order that they are added;

A. It is First in First Out // It is FIFO;

A. It is Last in Last Out // It is LILO;

MAX 1 [1]

1. (i) FrontPointer = 11

RearPointer = 52

QueueSize = 42

1 mark for all three values correct

(ii) FrontPointer = 11

RearPointer = 2

QueueSize = 44

1 mark for all three values correct [1]

A. incorrect value for FrontPointer if it matches the value given in part (b) (i) and incorrect value for QueueSize if it is equal to the value given for QueueSize in part (b) (i) incremented by two (follow through of errors previously made)

(iii) If DeckQueue is empty then

Report error

Else

Output DeckQueue[FrontPointer]

Decrement QueueSize

Increment FrontPointer

If FrontPointer > 52 Then

FrontPointer = 1

EndIf

EndIf [6]

**1 mark** for If statement to check if queue is empty – alternative for test is QueueSize = 0.

**1 mark** for reporting an error message if the queue is empty // dealing with the error in another sensible way – this mark can still be awarded if there is an error in the logic of the If statement, as long as there is an If statement with a clear purpose.

**1 mark** for only completing the rest of the algorithm if the queue is not empty – this mark can still be awarded if there is an error in the logic of the If statement, as long as there is an If statement with a clear purpose.

**1 mark** for outputting the card at the correct position

**1 mark** for incrementing FrontPointer and decrementing QueueSize

**1 mark** for If statement testing if the end of the queue has been reached

**1 mark** for setting FrontPointer back to 1 if this is the case – this mark can still be awarded if minor error in logic of If statement, eg >= instead of =

**A**. FrontPointer = (FrontPointer MOD 52) + 1 for 3 marks or FrontPointer = (FrontPointer MOD 52) for 2 marks, both as alternatives to incrementing and using and the second If statement - deduct 1 mark from either of the above if QueueSize has not been decremented

**A**. any type of brackets for array indexing I. Additional reasonable EndIf Statements

**MAX 5** unless all of the steps listed above are carried out

Chapter 38 – Lists

# In-text questions

**Q1:** A list can be implemented as an array.

**Q2:** (i) James, Paul, Sophie, Holly, Nathan, Tom

(ii) James, Paul, Sophie, Nathan, Tom

(iii) James, Melissa, Paul, Sophie, Nathan, Tom

**Q3:** The item could be appended to the end of the list, and then the list sorted.

Or, a linked list could be used.

**Q4:** The same pseudocode algorithm applies, but instead of inserting items in alphabetical order, the new item would be inserted at the end of the “subqueue” of items with the same priority.

**Q5:** It’s not a good idea to leave blank spaces in the list because then, functions like length(list) will not work correctly.

# Exercises

1. (a)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **List** | | | | |
| **ListLength** | **New** | **p** | **q** | **[1]** | **[2]** | **[3]** | **[4]** | **[5]** |
| 4 | 38 | - | - | 9 | 21 | 49 | 107 |  |
|  |  | 1 |  |  |  |  |  |  |
|  |  | 2 |  |  |  |  |  |  |
|  |  | 3 |  |  |  |  |  |  |
|  |  |  | 4 |  |  |  |  | 107 |
|  |  |  | 3 |  |  |  | 49 |  |
|  |  |  |  |  |  | 38 |  |  |
| 5 |  |  |  |  |  |  |  |  |

4,5 in sequence for ListLength;

1,2,3 in sequence for p;

4,3 in sequence for q;

Final list in array is 9, 21, 38, 49, 107;

**Do not award a mark if additional values indicated e.g. 4 for p** [4]

(b) Inserts an item/variable New into correct position/preserving order//into sorted list (or equivalent) [1]

(c) (i) Static structures have fixed (maximum) size whereas size of dynamic structures can change

// Size of static structure fixed at compile-time whereas size of dynamic structure can change at run-time;

Static structures can waste storage

space/memory if the number of data items stored is small relative to the size of the structure whereas dynamic structures only take up the amount of storage space required for the actual data;

Dynamic data structures (typically) require memory to store pointer(s) to the next item(s) which static structures (typically) do not need //

Static structures (typically) store data in consecutive memory locations, which dynamic data structures (typically) do not;

[1]

(ii) Memory allocated/deallocated at run-time/for new items (to dynamic data structure);

(Provides a) pool of free/unused/available memory;

**NE** to store new items

**MAX 1** [1]

Chapter 39 – Stacks

# In-text questions

**Q1:** SUB peek

IF isEmpty THEN

OUTPUT “Stack is empty“

ELSE

RETURN s(top)

ENDIF

ENDSUB

**Q2:** (i) [‘Blue’] stack pointer = 0

(ii) [empty] stack pointer = -1

(iii) [‘Yellow’] stack pointer = 0

# Exercises

1. (a) Stack [1]

(b) Overflow: attempting to add to stack which is full

*or*

Underflow: attempt to remove item from empty stack [2]

(c) Call stack – holding return addresses/parameters/local variables during execution of subroutines

*or* performing calculations

*or* holding IP addresses of web pages visited so that they can be viewed again by a user

*or* holding previous operations e.g. in a word processor, so they can be “undone” in reverse order. [2]

(d) For all elements in the queue/while queue is not empty

dequeue next element

place on stack

endfor

For all elements in stack/while stack is not empty:

pop element

add to queue

endfor [6]

Chapter 40 – Hash tables and dictionaries

# In-text questions

**Q1:** Item 34 has caused a collision and has been put in the next available free space.

**Q2:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **“Folded” value** | **Remainder** | **Location in hash table** |
| 123456 | 12+34+56=102 | 2 | 2 |
| 238464 | 23+84+64=171 | 71 | 71 |
| 188947 | 18+89+47=154 | 54 | 54 |
| 276084 | 27+60+84=171 | 71 | 72 (collision has occurred) |

**Q3:** (i) hash values:

BAG = (66 + 65 + 71) mod 11 = 202 mod11 = 4 (same as GAB)

TEA = (84 + 69 + 65) mod11 = 218 mod 11 = 9 (same as EAT)

(ii) Anagrams always hash to the same value

(iii) A weight could be given to each letter, so that BAG, for example is calculated as

(66 x 1 + 65 x 2 + 71 x 3) mod 11.

(Students may come up with many other suggestions!)

# Exercises

1. (a) A hash function is a function or algorithm which transforms a key into an address where the item may be stored. [1]

(b) Divide the StudentID by 1000 and take the remainder as the address at which the record is stored. *(or other suitable method)* [2]

2. (a) B6 [1]

(b) 421 [1]

(c) 500 (i.e. 0 to 499) [1]

(d) (i) A collision occurs when two items hash to the same address in the hash table

[1]

(ii) e.g. 1234567 and 4352567 (any 2 numbers which will result in the same remainder when divided by 500) [2]

(iii) The item can be put in the next available free space. (or other rehashing algorithm) [2]

Chapter 41 – Graphs

# In-text questions

**Q1:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Bury St Edmunds** | **Framling-ham** | **Wickham Market** | **Wood-bridge** | **Ipswich** | **Stow-market** |
| **Bury St Edmunds** |  | 57 |  | 56 | 45 | 25 |
| **Framlingham** | 57 |  | 10 |  | 31 |  |
| **Wickham Market** |  | 10 |  | 9 |  |  |
| **Woodbridge** | 56 |  | 9 |  | 15 |  |
| **Ipswich** | 45 | 31 |  | 15 |  | 21 |
| **Stowmarket** | 25 |  |  |  | 21 |  |

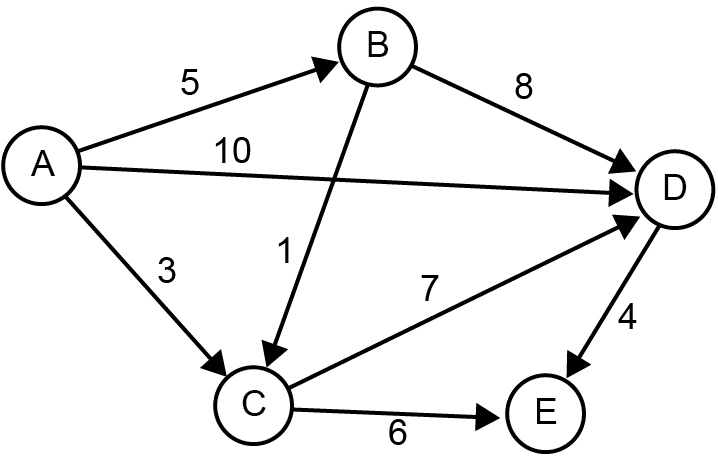
|  |  |  |  |
| --- | --- | --- | --- |
| A |  |  | **[B, C]** |
| B |  |  | **[A, C, D]** |
| C |  |  | **[A, B, F]** |
| D |  |  | **[B, E]** |
| E |  |  | **[D]** |
| F |  |  | **[C]** |

**Q2:**

# 

# Exercises

1. (a)



One mark for all edge and edge weight correct, one for arrows all correct [3]

|  |  |  |  |
| --- | --- | --- | --- |
| A |  |  | {B:5, C:3, D:10} |
| B |  |  | { C1:, D:8} |
| C |  |  | {D:7,E:6} |
| D |  |  | {E:4} |
| E |  |  | {} |

(b)

Three marks for all correct, deduct 1 for each error [3]

(c) Adding an edge is very simple/matrix is easy to work with. Appropriate for a highly connected graph

Adjacency list uses much less memory for a sparsely connected graph/ Easy to find all nodes connected to a particular node. List more appropriate for sparsely connected graph. [3]

2. This is an open question for students to research. A Google search on “practical applications of graph algorithms” will give many leads. See for example <http://www.graph-magics.com/practic_use.php>, or <https://www.quora.com/What-are-the-practical-industrial-oriented-applications-of-BFS-and-DFS>.

Chapter 42 Trees

# In-text questions

**Q1:** Leftmost subtree consists of Ben, Adrian, Steven and Sarah. The parent of Frank is Petra and Kate has two children, David and Ken.

There are 6 parent nodes – Louis, Ben, Anna, Kate, Adrian and Petra.

There are 12 child nodes (all except the root).

**Q2:** (iii) is not a treebecause it contains a cycle. All the others are trees

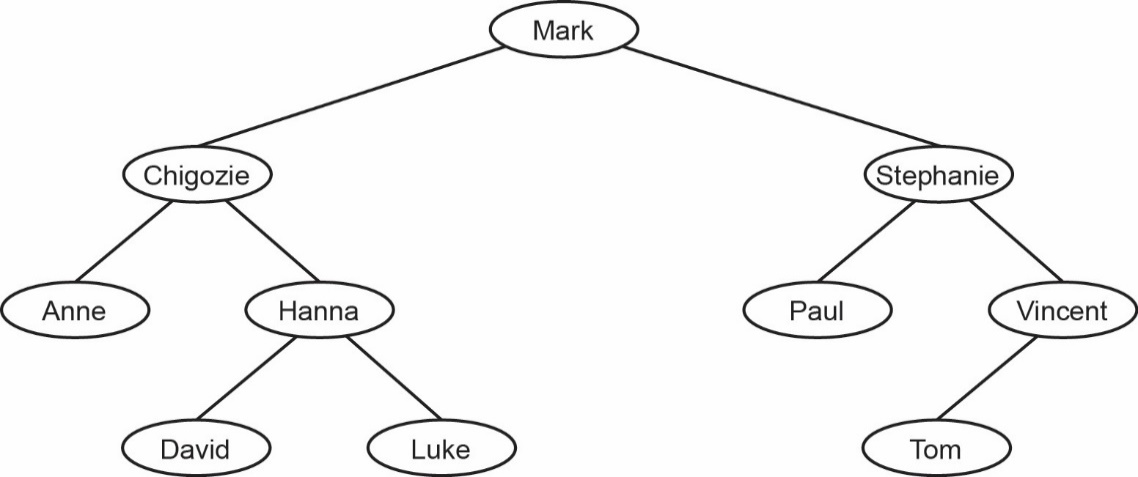
**Q3:** (a) 17, 8, 12, 14

(b) 17, 22,19

(c) 10 will have 12 as a parent, and be inserted to the left

20 will have 19 as a parent and be inserted to the right

**Q4**:



To find David, the names Mark, Chigiozie, Hanna and David would be checked.

**Q5:** The names would be output in alphabetical order -

Anne, Chigozie, David, Hanna, Luke, Mark, Paul, Stephanie, Tom, Vincent

**Q6:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **left** | **data** | **right** |
| tree[0] | 3 | Monkey | 1 |
| tree[1] | 2 | Topi | 5 |
| ,tree[2] | -1 | Ostrich | 8 |
| tree[3] | 6 | Giraffe | 4 |
| tree[4] | -1 | Hippo | 10 |
| tree[5] | -1 | Zebra | -1 |
| tree[6] | 9 | Buffalo | 7 |
| tree[7] | -1 | Cheetah | -1 |
| tree[8] | -1 | Rhino | -1 |
| tree[9] | -1 | Baboon | -1 |
| tree[10] | -1 | Jackal | -1 |

# Exercises

1. (a)

**1 mark** for all 5 lines correctly drawn

**1 mark** for all 5 arrowheads pointing in correct directions

**A** arrowheads at any position on line

**MAX 1** if more than 5 lines drawn by candidate (note that dotted arrow is given in question) [2]

(b) Adjacency matrix appropriate when there are many edges between vertices

// when edges may be frequently changed //

when presence/absence of specific edges needs to be tested (frequently)

Adjacency list appropriate when there are few edges between vertices

// when graph is sparse // when edges rarely changed

//when presence/absence of specific edges does not need to be tested (frequently)

**A**(Accept) alternative words which describe edge e.g. connection, line [2]

(c) Undirected // No direction is associated with each edge;

Has no cycles

// No (simple) circuits

// No closed chains

// No closed paths in which all the edges are different and all the intermediate vertices are different

// No route from a vertex back to itself that doesn’t use an edge more than once or visit an intermediate vertex more than once;

MAX 2 [2]

Alternative definitions:

Graph with no cycles, and a simple cycle is formed if any edge is added to it;

Graph which is connected, and it is not connected anymore if any edge is removed from it;

Graph in which any two vertices can be connected by a unique simple path; (Note: not just adjacent vertices)

Graph which is connected and has n − 1 edges where n is no of vertices;

Graph which has no simple cycles and has n − 1 edges where n is no of vertices;

(d)

Jack

Bramble

Snowy

Squeak

Pip

Butter

Bear

(e) 1 mark for Bramble and Snowy as children of Jack

1 mark for four correct children of Bramble and Snowy

DPT if arrows drawn instead of lines

DPT if any node has more than 2 child nodes

Accept "mirror image" answers which are consistent. [3]

(e) For solution with 3 arrays:

One array stores data items;

One array for left child pointers;

One array for right child pointers;

Pointers stored at same location in arrays as corresponding data item;

For solution with 1 array of records:

Record created to store data item and pointers;

One pointer to left child;

One pointer to right child;

For either of the above solutions:

Rogue value (allow example) used to indicate no child;

Variable indicates position in array(s) of root node

// Root node stored at first location/start of array(s);

If answered as diagram:

Column for data with at least three correct data items in it;

Use of rogue value for a node that does not have child;

Correct value for a start pointer variable indicating position of root node in the array (not drawn as an arrow, array indices must be labelled);

Column for left child pointers\*;

Column for right child pointers\*;

\* = To get these marks, there must be a sufficient number of pointers to demonstrate that the data structure is a representation of a binary tree, but it is not necessary for every item to be shown. Also the array indices must be shown.

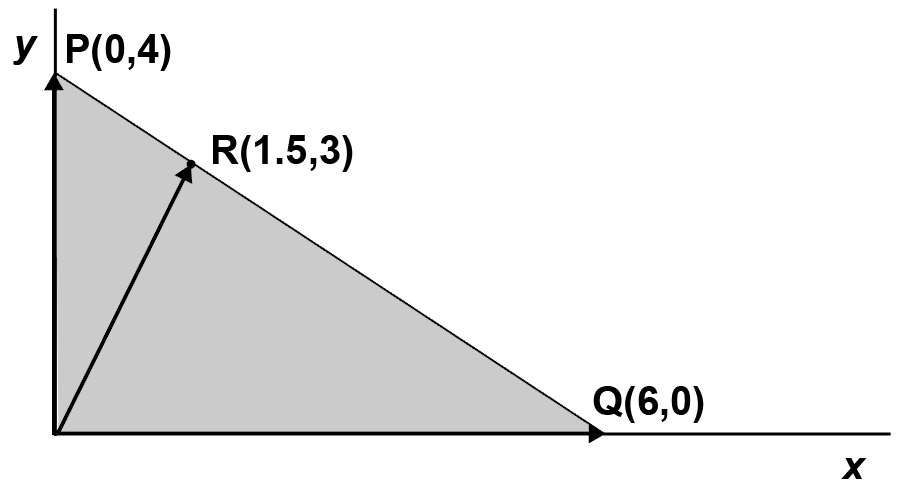
MAX 3 [3]

Chapter 43 - Vectors

# In-text questions

**Q1:** a = sqrt(22 + 32) using Pythagoras’ theorem. c = sqrt(6.52 + 42)

**Q2:**



**Q3:** cos θ = a • b

||a|| • ||b||

= (3 x 6) + (4 x 0)

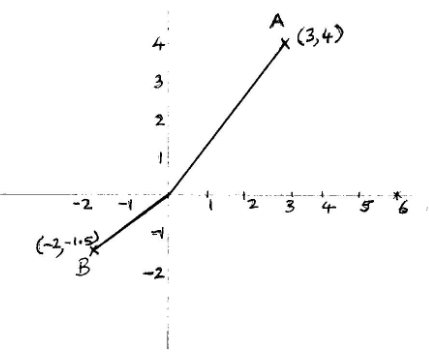
5 x 6

= 18

30

= 0.6 **θ = 53°**

**Q4:** Dot product = [3,4] • [-2,-1.5]  
 = (3 x -2) + (4 x -1.5)  
 = -6 – 6  
 = **-12**

 (a) No, they are not at right angles. See illustration.

(b) cos θ = (3 x 7.5) + (4 x 10)

5 x **√** 56.25 + 100

= 22.5 + 40

5 x 12.5

= 62.5

62.5

= **1**

(c) θ = 0

# Exercises

1. (a) (11, 7) [1]

(b) 10 + 12 = 22 [1]

(c) (3, 9) [1]

2. (a) 10 is printed. The algorithm calculates the dot product of the two vectors. [3]

(b) (i) The length of the vector from the origin to A = sqrt(42 + 32) = 5 (Pythagoras)

B is therefore 1/5 the length of A so

(x,y) is (4/5, 3/5) or (0.8, 0.6)

(Check: 9/25 + 16/25 = 25/25 = 1)

[2]

(ii) The parameter A is a list or array of 2 elements which are the x and y coordinates of the vector.

SUB normalise(A)

normalA 🡨 [0,0]

length 🡨 sqrt(A[0]\*\*2 +A[1]]\*\*2)

normalA[0] 🡨 A[0]/length

normalA[1] 🡨 A[1]/length

RETURN normalA

ENDSUB

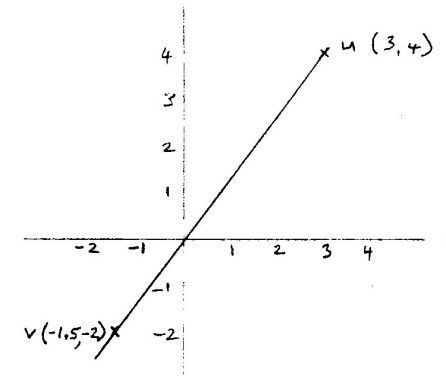
[4]

3. (a) u = (3,4) v = (-1.5,-2)

u • v = (3 x -1.5) + (4 x -2)  
 = -4.5 + -8  
 = **-12.5**

(b) ||u||2 = 32 + 42

= 25  
||u|| = 5  
||v||2 = -1.52 + 22  
 = 2.25 + 4  
 = 6.25

 ||v|| = 2.5

cos θ = -12.5

5 x 2.5

= -12.5

12.5

= -1

(c) θ = 180° See illustration. [3]

Chapter 44 – Recursive algorithms

# In-text questions

**Q1:** The OUTPUT statement is executed 4 times, as the RETURN statement is reached 4 times

1

2

6

24

*(see program factorial.py in folder)*

**Q2:** Pseudocode:

#iterative routine

SUB sumList(numList)

sum 🡨 0

FOR i in numList

sum 🡨 sum + i

ENDFOR

RETURN sum

ENDSUB

#recursive routine

SUB sumListRecursively(numList):

IF len(numList) = 1 THEN

RETURN numList[0]

ELSE

RETURN numList[0] + sumListRecursively(tail of numList))

ENDSUB

#main program

numbers 🡨 [3, 5, 10, 2]

total 🡨 sumList(numbers)

OUTPUT('Sum =',total)

total 🡨 sumListRecursive(numbers)

OUTPUT('Using a recursive routine:')

OUTPUT('Sum =',total)

Python programs:

#iterative routine

SUB sumList(numList)

sum 🡨 0

FOR i in numList

sum 🡨 sum + i

ENDFOR

RETURN sum

ENDSUB

#recursive routine

SUB sumListRecursive(numList)

IF len(numList) = 1 THEN

RETURN numList[0]

ELSE

RETURN numList[0] + sumListRecursive(numList[1:])

#main program

numbers 🡨 [3, 5, 10, 2]

total 🡨 sumList(numbers)

OUTPUT('Sum =',total)

total 🡨 sumListRecursive(numbers)

OUTPUT('Using a recursive routine:')

OUTPUT('Sum =',total)

*(see program sum list.py in folder)*

# Exercises

1. (a) Any three at one mark each

Procedure calls itself;

Base case enables escape from recursion;

System keeps pending calls on a stack;

Parameters also passed on a stack;

Need a stack;

System needs a lot of memory to handle recursion;

Recursive programs are usually shorter than their non-recursive equivalents; [3]

(b) (i)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **numlist** | | | |  |  |
| **Length (numlist)** | **0** | **1** | **2** | **3** | **first** | **new** |
| 4 | 3 | 5 | 10 | 2 | 3 |  |
| 3 | 5 | 10 | 2 |  | 5 |  |
| 2 | 10 | 2 |  |  | 10 |  |
| 1 | 2 |  |  |  | 2 |  |
| 0 | none |  |  |  |  |  |
| 1 | 2 |  |  |  |  |  |
| 2 | 2 | 10 |  |  |  |  |
| 3 | 2 | 10 | 5 |  |  |  |
| 4 | 2 | 10 | 5 | 3 |  | [2, 10, 5,3] |

1 mark for each correct row up to 6 marks [6]

(ii) It reverses the list. **new** will be set to [2, 10, 5, 3] [1]

2. (a) The subroutine is called 4 times;

Call 1: it is called from the main program with parameter 4, return address line 8.

Call 2: it is called at line 5 with parameter 3, return address line 6

Call 3: it is called at line 5 with parameter 2, return address line 6

Call 4: it is called at line 5 with parameter 1, return address line 6

Now it starts to “unwind”. It takes the top address off the stack (line 6) and returns

stars(1) return value \*

stars(2) + \* return value \*\*

stars(3) + \* return value \*\*\*

stars(4) + \* return value \*\*\*\*

Next return address is line 8 where it prints \*\*\*\* [4]

(b) \*\*\*\* [1]

(c) \*234 [1]

*(see Python program recursive stars.py*

Chapter 45 – Big-O notation

# In-text questions

**Q1:** The second one is more efficient because for any value of n it only performs one operation. The other algorithm performs n + 1 operations.

**Q2:**

|  |  |
| --- | --- |
| **x** | **y = 2x** |
| 1 | 2 |
| 10 | 1,024 |
| 100 | 1,267,650,600,228,229,401,496,703,205,376 |

*2100 = A very large number!*

**Q3:** There are 4 x 3 x 2 x 1 = 24 permutations of 4 objects.

There are 6 x 5 x 4 x 3 x 2 x 1 = 720 ways of arranging six students in a line.

**Q4:** The first method cannot use a binary search as the hacker does not know what word he is looking for. His algorithm has to do a linear search of 170,000 items to check the lower case letters.

Using the second method, each letter could be one of (26 + 26) different characters. The hacker has to check 526 combinations which is vastly more.

Mixing uppercase and lowercase letters and including numbers and special symbols makes a password much more difficult to hack.

**Q5:** The first algorithm has time complexity O(n) and the second one, O(1) (i.e. constant for any value of n.

# Exercises

1. The algorithm performs 1 initialisation statement, followed by two nested loops.

The outer loop is performed (n-1) times.

The innermost loop is performed (n-1) + (n-2) …1 times

This is a total number of statements of around ½ (n-1)2

Although this is less than n2, it is nevertheless of that order. Therefore, the time complexity is O(n2).

(An acceptable answer would be, there are two nested loop each performed a maximum of n (or more accurately n-1) times, so complexity is O(n2).)

Give 1 mark for correct answer, 2 for explanation.

See Python program **Chapter 45 find duplicate.py** and **Chapter 45 find duplicate 10000 items.py**

2. (a)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **1** | **2** | **4** | **8** | **12** |
| **f(n) = n2** | 1 | 4 | 16 | 64 | 144 |
| **f(n) = 2n** | 2 | 4 | 16 | 256 | 4096 |
| **f(n) = log2 n** | 0 | 1 | 2 | 3 | 3.585 |
| **f(n) = n!** | 0 | 2 | 24 | 40320 | 479,001,600 |

[4]

(b) C, A, D, B, E [2]

(c) For even a relatively small value of n, e.g. 20, it would take thousands of years to compute a solution on a fast computer. For n = 25, it would take billions of years.

If n <= 4 (or maybe 5?), such an algorithm might be considered suitable. [3]

(d) The merge sort is more efficient.

For 1024 items, n log2 n = 1024 x 10, and n2 = 1024 x 1024 which is much greater, with the difference becoming greater the larger the data set becomes.

*(or any other suitable values)* [3]

Chapter 46 – Searching and sorting

# In-text questions

**Q1:** Maximum one million. Average half a million.

**Q2:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **itemSought** | **index** | **found** | **lower** | **upper** | **midpoint** | **aList(midpoint)** |
| 90 | -1 | False | 0 | 11 | 5 | 35 |
|  |  |  | 6 |  | 8 | 86 |
|  |  |  | 9 |  | 10 | 93 |
|  |  |  | 9 | 9 | 9 | 90 |
|  | 9 | True |  |  |  |  |

**Q3:** 219 = 524,288**and** 220 = 1,048,576 therefore 20 items will need to be examined.

In a dataset of between 2n and 2n+1 -1 items, you need to examine a maximum of n + 1 items 2n items, a maximum of n+1 items have to be examined.

Looking for the last item in the list requires examining the maximum.

          If you try it with a dataset of 7 items, looking for the last item, 3 items (indexed 0-6) are examined. (3, 5 and 6) With 8 (23) items, items 3, 5, 6 and 7 are examined.

**Q4:** Upper < lower i.e. every item in the list has been searched, -1 is returned

OR

alist[midpoint] = itemSought i.e. item found, index is returned

In either case, a value is returned to the calling program before the end of the function is reached.

*Binary search (iterative) – see program Chapter 46 Binary Search (Iterative).py*

*Binary search (recursive) – see program Chapter 46 Binary Search (recursive).py*

*Binary search tree – see program Chapter 46 Binary Search Tree.py*

**Q5:** (b) [54] and [36].

*Merge sort – see program Chapter 46 merge sort.py*

**Q6:** Note: An odd number of initial sequence values is split using integer division.

# 

# Exercises

1. The number of records to be sorted (1)

whether the computer has sufficient memory to hold all the records at once (1)

whether the records are already partially sorted (1) [2]

2. (a)

|  |  |  |
| --- | --- | --- |
| **Position** | **Value** | **Order examined in** |
| 8 | Philip | 1 |
| 10 | Ravi | 3 |
| 11 | Richard | 4 |
| 12 | Timothy | 2 |

1 mark for row 8 correct

1 mark for row 11 correct

1 mark for both rows 10 and 12 correct [3]

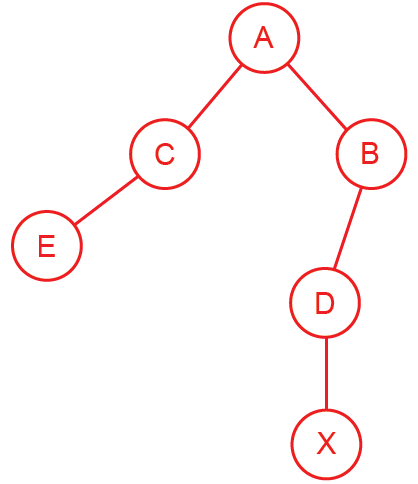
(b) 8 (27 = 128, 28 = 256) [1]

(c) O(log2 n) [1]

Chapter 47 – Graph traversal algorithms

# In-text questions

**Q1:** (a)

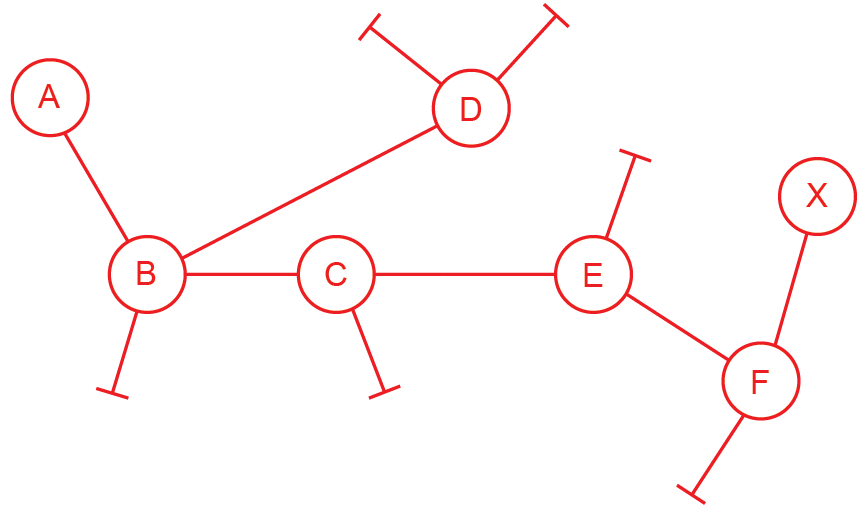


(b) It is a connected undirected graph; with no cycles.

(c)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **X** |
| **A** |  | 1 | 1 |  |  |  |
| **B** | 1 |  |  | 1 |  |  |
| **C** | 1 |  |  |  | 1 |  |
| **D** |  | 1 |  |  |  | 1 |
| **E** |  |  | 1 |  |  |  |
| **X** |  |  |  | 1 |  |  |

**Q2:**



# Exercises

1. (a) (i) Stack [1]

(ii) Queue [1]

(b) (i) A B C G D E F H [3]

(ii) A B D E C G F H [3]

(c) (i) The tree is not a graph because it has connected cycles. The edges A to D and D to E would need to be removed for it to be a tree.

(ii) Using a pre-order traversal, nodes are visited in the order 1 2 4 5 7 3 6 8 9

This is the same order that they would be traversed using a depth first traversal.

*See programs in folder: Chapter 47 dfs.py*

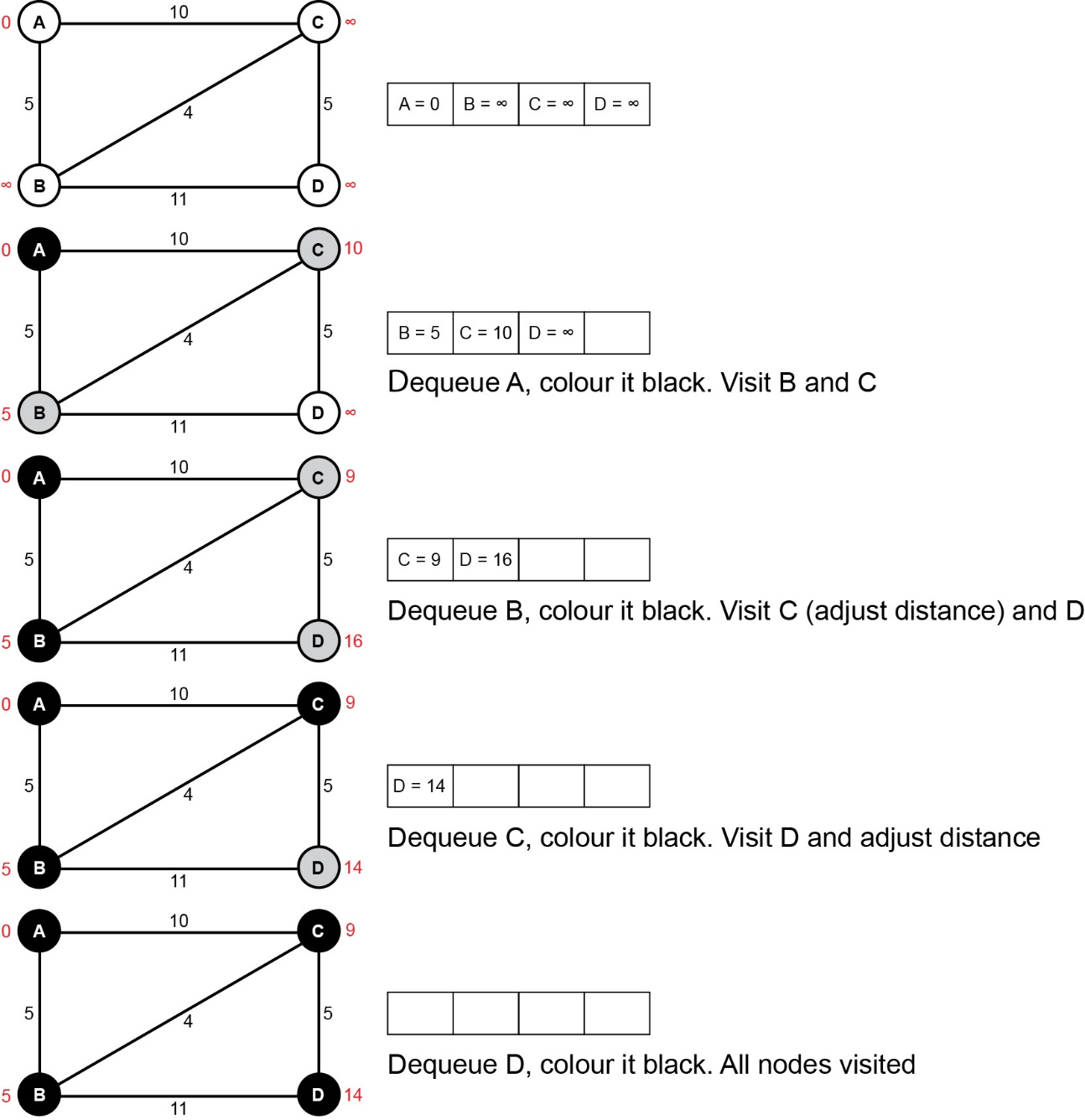
*Chapter 47 bfs.py*

*Chapter 47 bfs iterative version.py*

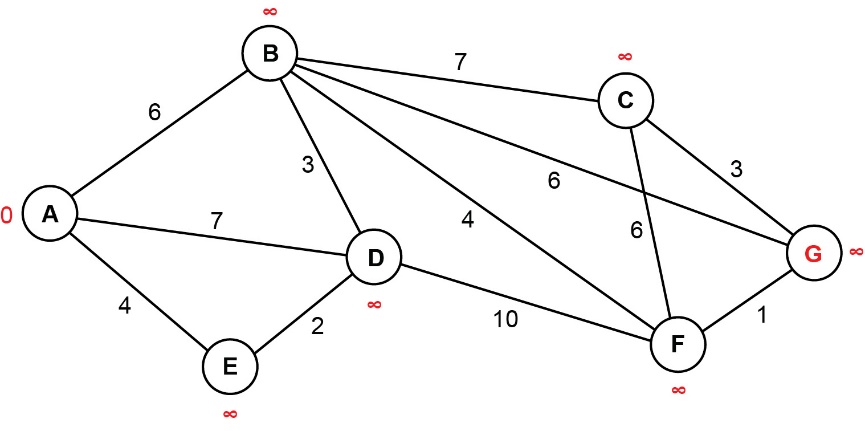
Chapter 48 – Optimisation algorithms

# In-text questions

**Q1:** The steps are shown below. It is not necessary to draw the graph for each step – the “temporary distances” can be crossed out and replaced with a lower value when necessary. In this simple example, the priority queue does not need adjusting as all distances remain in the correct ascending order as they are adjusted.

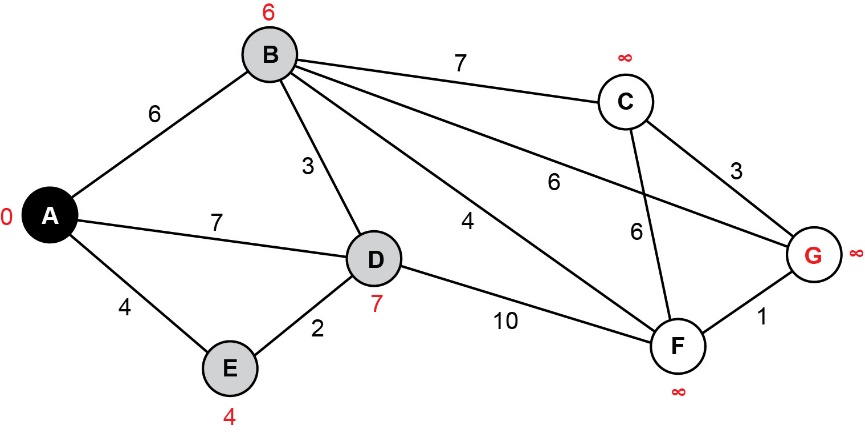


**Q2:** The steps are as follows:



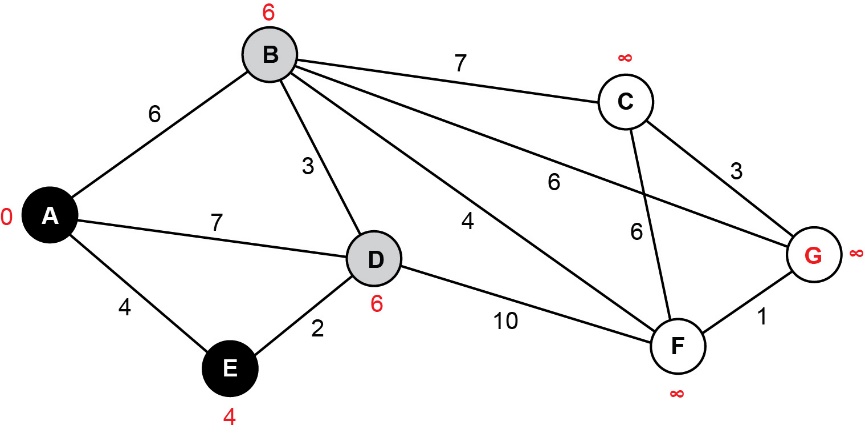
Add all vertices to priority queue. Set distance of A to 0, all other nodes to ∞.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A = 0 | B = ∞ | C = ∞ | D = ∞ | E = ∞ | F = ∞ | G = ∞ |

****

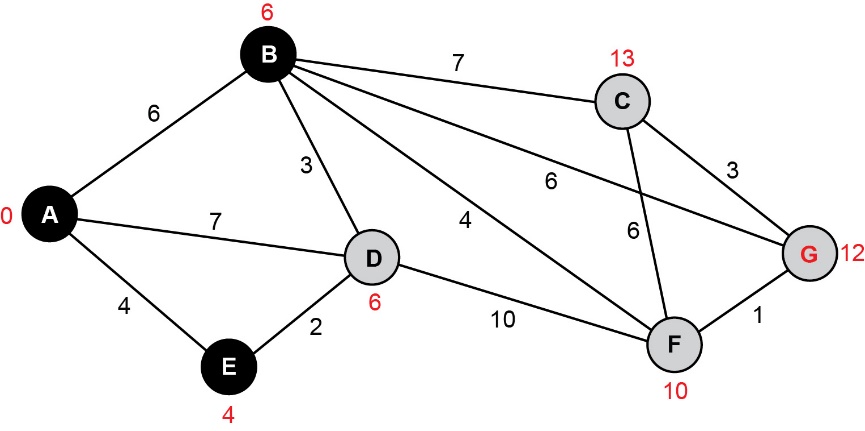
Add all vertices to priority queue. Set distance of A to 0, all other nodes to ∞.  
Dequeue A and shade black. Visit neighbours of A, adjust temporary distances.   
Colour E,D,B grey. Adjust priority queue.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E = 4 | B = 6 | D = 7 | C = ∞ | F = ∞ | G = ∞ |  |

****

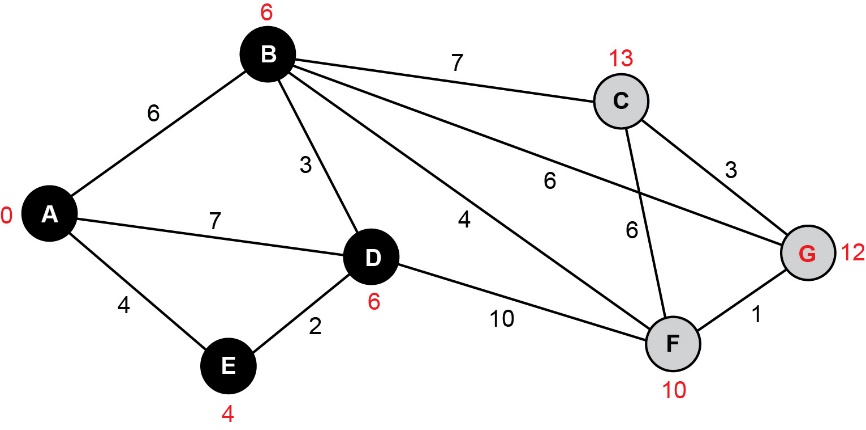
Dequeue E. Visit neighbours of E, colour D grey, adjust temporary distances.   
No need to adjust priority queue.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B = 6 | D = 6 | C = ∞ | F = ∞ | G = ∞ |  |  |

****

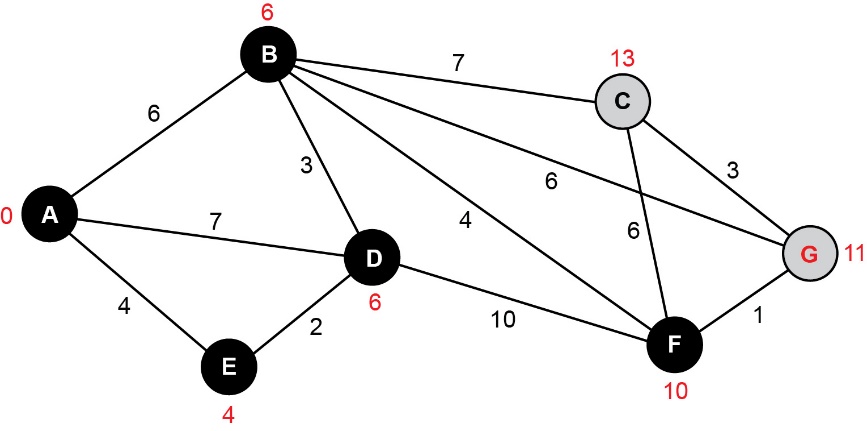
Dequeue B. Visit neighbours of B, colour them grey, adjust temporary distances.   
Adjust priority queue.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D = 6 | F = 10 | G = 12 | C = 13 |  |  |  |

****

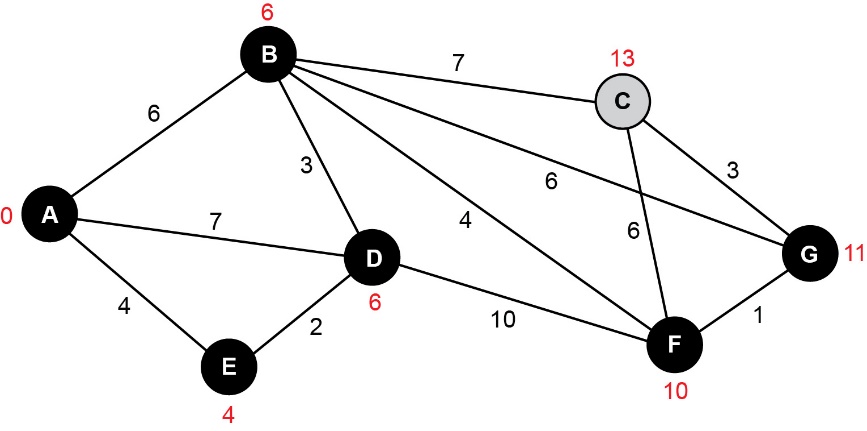
Remove D from queue.   
Neighbours of D have already been visited, and no adjustments to distances are needed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F = 10 | G = 12 | C = 13 |  |  |  |  |

****

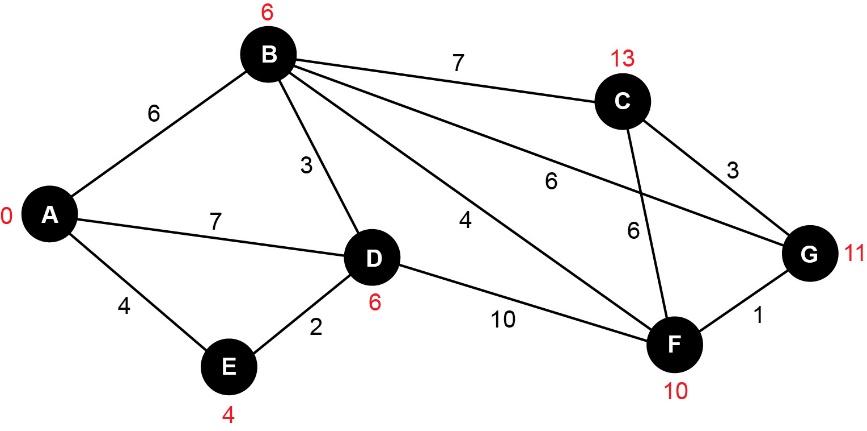
Dequeue F. Adjust temporary distances.  
Neighbours of F have already been visited.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| G = 11 | C = 13 |  |  |  |  |  |

****

Remove G from queue. Visit neighbours of G, no adjustments to distances are needed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C = 13 |  |  |  |  |  |  |

****

Remove C from queue. All nodes have now been visited.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

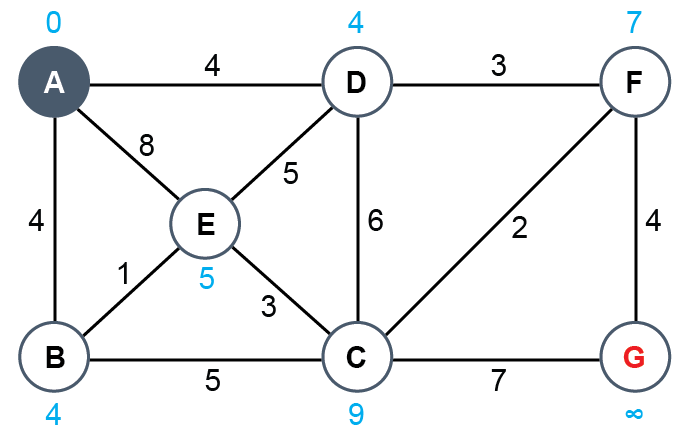
**The shortest distance from A to G is 11.**

# Exercises

1. (a) Djikstra’s shortest path algorithm is designed to find the shortest path from source to every other node in a weighted graph. [2]

(b) Used in traffic routing problems, e.g. Google Maps to find the shortest route from A to B. Also used in network communications to find the best route across the Internet from one computer to another. [4]

(c)



½ mark for correct answers at A, B, C, D, F, G. 1 mark for correct answer at E.

2. (a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Liverpool** | **Manchester** | **Leeds** | **Sheffield** | **York** |
| 0 | 34 | 75 | 74 | ∞ |

[3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Liverpool** | **Manchester** | **Leeds** | **Sheffield** | **York** |
| 0 | 34 | 75 | 73 | 103 |

(b)

[2]

[2]

Chapter 49 – Limits of computation

# In-text questions

**Q1:** (a) 9! = 362,880

(b) 19! = 121645100408832000

(c) 49! = 8281864034267560872252163321295376887552831379210240000000000

**Q2:** If n = 2, for example, O(2n) = 4 which is the same as O(n2), the polynomial time solution.

**Q3:** A is tractable B and C are intractable. They do not have polynomial time solutions. They may be computable in a relatively short time for very small values of n.

**Q4:** Artificial intelligence for example is a whole field of mostly incomputable problems - anything to do with human behaviour in fact. Could we compute whether the accused in the dock is guilty? Lie detectors are unreliable.

Any problem which does not have an optimal solution but is a "trade-off" between several factors, marketing problems and product design, beauty vs utility vs price being examples.

How reliable are the solutions produced by "Big Data" techniques?

# Exercises

1. (a) (i) O(an) Accept Exponential, an [1]

(i) A [1]

(b) (i) The problem can be solved // algorithm exists for problem;

But it cannot be solved in polynomial time // but not quickly enough to be useful;

It takes an unreasonable amount of time; to solve [2]

(ii) Use of heuristic; algorithm that makes a guess based on

experience;

That provides a close-to-optimal solution/approximation; that

only works in some cases; **A** non-optimal

Example of heuristic method e.g. hill-climbing/stochastic/local

improvement/greedy algorithms/simulated annealing/trial and

error/any reasonable example;

Relax some of the constraints on the solution; **A** solve simpler

version of problem

**MAX 2** [2]

2. (a) Is it possible in general to ***write a program/algorithm*** ***// create a Turing machine*** that can tell, given any program and its inputs and without ***running/executing the program*** whether the given program with the given inputs will halt?

**A** “it” in second reference to program.

**A** “create a Turing machine” for “write an algorithm” [2]

(b) Shows that some problems are non-computable / undecideable // shows that some problems cannot be solved by a computer/algorithm; [2]

In general, inspection alone cannot always determine whether any given algorithm will halt for its given inputs // a program cannot be written that can determine whether any given algorithm will halt for its given inputs. [1]

**A** it is not computable

**MAX 1**

Chapter 50 – Mealy machines

# In-text questions

**Q1:** A packet of crisps is dispensed and no change is given.

**Q2:** Input = 1 0 1 1 0 1

State sequence = S0 S2 S1 S2 S2 S1 S2

Output = n n n y n n

The final state is S2 and n is the final output.

**Q3:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Current state** | **Output** | **Next state** |
| 0 | S0 | 0 | S1 |
| 1 | S0 | 0 | S2 |
| 0 | S1 | 0 | S1 |
| 1 | S1 | 1 | S2 |
| 0 | S2 | 1 | S1 |
| 1 | S2 | 0 | S2 |

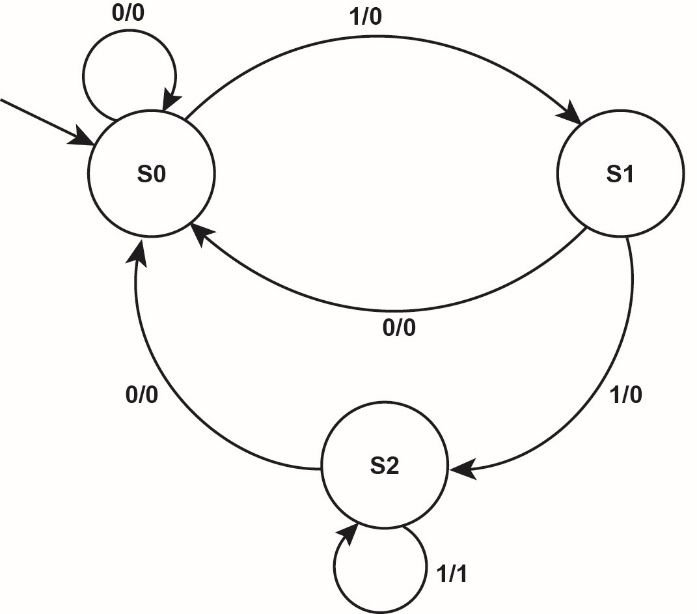
**Q4:** Input = 0 0 1 0 0 1

State sequence = S0 S1 S1 S2 S1 S1 S2

Output = 0 0 1 1 0 1

# Exercises

1. (a)



(b) (i) Input = 1 1 0 1 1 1

State sequence = S0 S1 S2 S0 S1 S2 S2

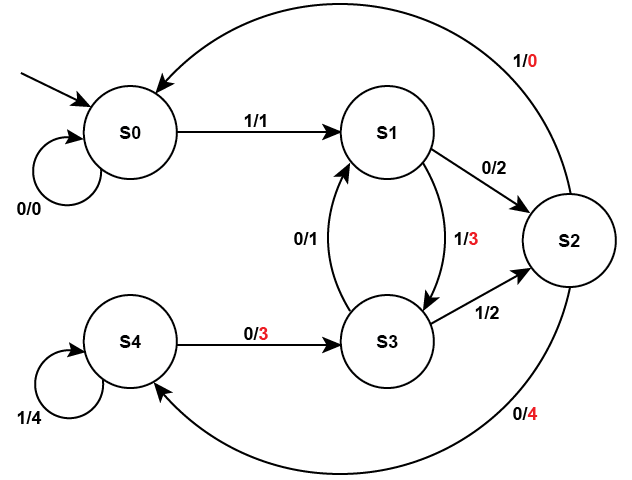
Output = 0 0 0 0 0 1

(ii) Input = 1 0 1 1 0 1

State sequence = S0 S1 S0 S1 S2 S0 S1

Output = 0 0 0 0 0 0

2.



(a) A = 3, B = 3, C = 0, D = 4

(b) Input 1 0 1 0 0 1 1

State S0 S1 S2 S0 S0 S0 S1 S3

Output 1 2 0 0 0 1 3

(c) 3 (which is the remainder when 83, or 1010011 in binary, is divided by 5.)

Chapter 51 – Sets

# In-text questions

**Q1:** A = {2, 3, 5, 7, 11, 13, 17, 19}

**Q2:** B = {2, 4, 6, …}

**Q3:** “A is the set consisting of those elements x such that “x ∊ N and x ≥ 1 is true”

**(**or alternatively but not in quite the given words, “A is the set consisting of all integers greater than 0”.)

**Q4:** A = { or A = {

**Q5:** 2, 4, 6, 8

**Q6:** (a)A = {5x | x ∊ N ∧ x ≥ 1} using set comprehension

(b) B = { x | x ∊ N ∧ x ≥ 1 ∧ x ≤ 9}

(c) C = {2x | x ∊ N ∧ x ≥ 1 }

**Q7:** {(4,8), (8,8), (3,8)}

**Q8:** None of them are true.

**Q9:** A = {1, 3, 5}

**Q10:** A ∩ B = ∅

**Q11:** C = {1, 9, 11} (Note that 1 is not a prime number)

# Exercises

1. A = {0, 1}

2. (a) (i) A ⊆ B means A is a subset of B. e.g. A = {1, 3, 5} B = {1, 2, 3, 4, 5, 6}

(ii) A ⊇ B means A is a superset of B. e.g. A = {0, 2, 4, 8, 16} B = {4, 8}

(b) A ⊇ B is true. A = {2, 4, 6, …} B = {4, 8, 12, …} so A is a superset of B

3. (a) (i) S2, S5 and S6

(ii) S4

(b) {(1,3), (1,1), (2,3), (2,1), (3,3), (3,1), (4,3), (4,1)}

(could be in a different order)

(c) S6 = S1 \ S2 or S6 = S4 \ S2

(d) (i) S7 = {1, 2, 3, 4, 5}

(ii) S8 = {3, 4}

Chapter 52 – Regular expressions

# In-text questions

**Q1:** R = colou?r or col(o|ou)r

**Q2:** a\*b+

**Q3:** Scooby, Scoby, Scobby, Scoobdbdbdy

**Q4:** ab\*(a|b)

**Q5:** a(ba)\*a

# Exercises

1. (a) ab+c *or* abb\*c *or* ab\*bc [1]

(b) (0|1)1\* or (1|0)1\* or 0 |(0?1\*)

2. (a) (i) One or more a’s followed by b [1]

(ii) The strings of ab or b // zero or one a’s followed by b [1]

(iii) A sequence of zero or more occurrences of ab [1]

(b) (i) Clai?re or Clare|Claire or Cla(ir|r)e or other valid possibilities [1]

(ii) 10(0|1)\*01 [2]

**1 mark** for the 10 at the start and 01 at the end

**1 mark** for (0|1)\* in the middle to produce a correct expression

**A** use of different types of brackets

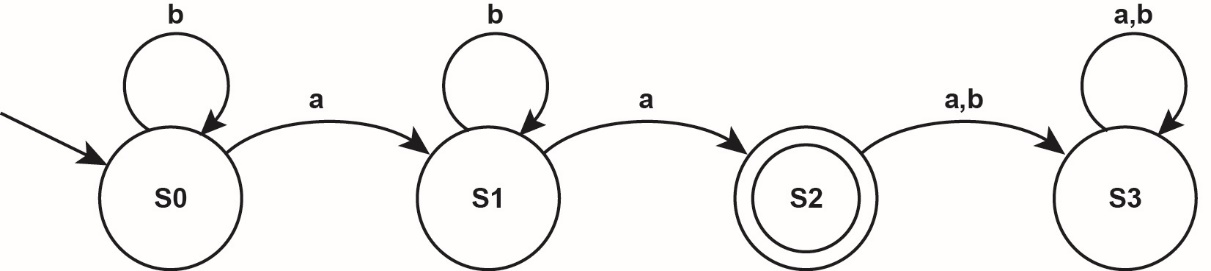
Award **2 marks** for any other expression that would work

3. (a) 11001, 101111 [3]

*(2 marks for 2 correct, plus one mark for not including any incorrect strings)*

(b) 0\*1+(1\*(00 | 01)\*)\*1\* [3]

4. See below:



[3]

Chapter 53 – Turing machines

# In-text questions

**Q1:**

S2

1

1

0

0

1

1

1

0

1

1

0

1

1

1

0

0

1

0

S0

S1

S0

S0

S1

S2

S1

0

1

1

0

0

S2

0

S3

S2

**Q2:** δ (S0, 0) = (S0, 0, R)

δ (S0, 1) = (S0, 1, R)

δ (S0, ) = (S0, , L)

# Exercises

1. (a)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 1 | 1 | 1 | 1 |  |  |  | . . . . | Current state | S1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 1 | 1 |  |  |  |  |  | S1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 1 | 1 |  |  |  |  |  | S1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 1 | 1 |  |  |  |  |  | S1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 1 | 1 |  |  |  |  |  | S2 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 1 |  |  |  |  |  |  | S3 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 | 1 |  |  |  |  |  |  |  | S4 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

1 mark for each of the top 5 rows

1 mark for the sixth and seventh row together

Must have correct tape contents and state for each mark

If the read/write head is missing, deduct mark first time it is missing only. [6]

(b) Deletes two ones/bits from the right hand end of the string (NOT end of tape)

***OR***

subtracts two from a unary number [1]

(c) A Turing machine that can execute/simulate the behaviour of any other Turing machine //can compute any computable sequence.

Faithfully executes operations on the data precisely as the simulated TM does

Description of the / instructions for TM and the TM’s input are stored on the UTM’s tape //the UTM acts as an interpreter.

Alternative definition:

A UTM, U is an interpreter that reads the description <M> of any arbitrary Turing machine M and faithfully executes operations on data D precisely as M does. The description <M> is written at the beginning of the tape,   
followed by D. [2]

***“Turing’s paper… contains, in essence, the invention of the modern computer and some of the programming techniques that accompanied it.” (Minsky (1967)***

Chapter 54 – Backus-Naur form

# In-text questions

**Q1:** Peter and Anna may or may not be married to each other

There may be one or several salesmen involved

Either “Close one eye and look at that dog” or “Look at that dog who has only one eye”

**Q2:** <positive integer> ::= <non-zero digit> | <positive integer><digit>

< digit> ::= <non-zero digit>|<0>

<non-zero digit> ::= 1|2|3|4|5|6|7|8|9

**Q3:** <real number> ::= <integer> <point> <integer>

< integer> ::= <digit> | digit> <integer>

<digit> ::= 1|2|3|4|5|6|7|8|9|0

<point> := .

**Q4:** <expression> ::= <term>|<expression> + <term>|<expression> - <term>

<term> ::= <variable>|<term> \* <variable>|<term> / <variable>

<variable> ::= a|b|c|d

(a) a – b

a is a <variable> and is therefore a <term>, and hence an <expression>

b is a <variable> and is therefore a <term>

a - b is <expression> - <term> and therefore an <expression>.

(b) a + b \* c

b is a <variable> and is therefore a <term>

c is a <variable>

b \* c is <term> \* <variable> and therefore a <term>

a is <variable> and is therefore a <term>, and hence an <expression>

a + b \* c is <expression> + <term> and therefore an <expression>

(c) As above, a \* b is a term, c is a <variable>, therefore

a \* b \* c is< term> \* <variable> which is a <term> and therefore an <expression>.

# Exercises

1. (a) 4686: 4 is a <digit>, therefore a <value>.

Applying the recursive rule for <value>., 4686 is a <value>.

7 + 8: <digit> + <digit>, therefore 7 + 8 is a <sum>.

05 + 170: 0 is a <digit>, therefore a <value>

Applying the recursive rule for <value>, 05 is a <value>

Applying the same rules, 170 is a <value>, therefore 05 + 170 is a <sum>

(b) <hex> ::= <digit>|<letter>|<hex><digit>|<hex><letter>

<digit> ::= 1|2|3|4|5|6|7|8|9|0

<letter> ::= A|B|C|D|E|F

(c)

value:

digit

digit:

1

2

3

4

5

6

7

8

9

0

2. (a) aea is valid, since e is a vowel, and aea is a<vowel>a therefore a vowel string.

uuu is valid, since u is a vowel, and uuu is u<vowel>u therefore a vowel string.

A and E are undefined therefore AEA is not a valid vowel string

aeae is not a valid vowel string as it contains four vowels and only three are allowed

oooaeio is invalid since although a,e,i,o are all in <vowel>, ooaeio is not a valid vowel-string as it contains more than three vowels.

(b) <vowel-string> ::= <vowel>| double-vowel |a<vowel-string>a |e< vowel-string >e | i<vowel-string>i | o<vowel-string >o | u< vowel-string >u

vowel ::= a|e|i|o|u

<double-vowel> ::= aa|ee|ii|oo|uu

3. **AC 234** is valid, since

AC is a valid code,

234 is <pos digit><digit><digit> and therefore a valid number,

AC 234 is <code><’ ‘><number> therefore a valid reg-no.

**AB 13** is not valid since

AB is not a valid code therefore this is not a valid reg-no.

**AX 099** is not valid since 099 is not a valid number (does not start with pos digit.)

**BB 2345** is valid since

BB is a valid code

2340 is <pos digit><digit><digit><digit> and therefore a valid number,

BB 2340 is <code><’ ‘><number> therefore a valid reg-no.

**AX 6** is not valid since 6 is not a valid number.

Chapter 55 - Reverse Polish notation

# In-text questions

**Q1:** get b

get c

multiply them

**Q2:** get b

get c

get d

add c and d and store the intermediate result

multiply b by the intermediate result of (c + d)

*or possibly*

get c

get d

add them and store the intermediate result

get b

multiply b by the intermediate result of (c + d)

**Q3:** (a)

1 3 2 9 4 6 5 8 7

( a + b ) - x ^ y \* 3

Reverse Polish form is a b + x y ^ 3 \* -

(b) 1 10 3 2 9 4 6 5 8 7

x = - a + ( c - d ) / e

Reverse Polish form is x a – c d – e / + =

**Q4:** (6 + 3) \* (7 – 2)

**Q5:** 1 3 2 5 4 9 6 8 7

( 5 + 9 ) / 2 - ( 2 \* 3 )

Reverse Polish form is 5 9 + 2 / 2 3 \* -

5

57

14

14

7

7

7

7

1

9

2

2

2

6

3

**Q6:** It is good practice for students to create binary expression trees to aid understanding and help find the result of a traversal if given a tree. This is more of a discussion question.

To construct the binary tree, start with a fully parenthesized expression and decide which is the “main” operator which will finally make up the whole expression. This goes at the root (\* in the example given).

Then for each parenthesised expression to the left of the main operator, create a subtree in the same way. Repeat for the right subtree.

\*

+

a

b

-

c

d

Post-order traversal: a b + c d - \*

Note: The first tree representing the expression (7 + 10 / 5) + (6 \* 2) is constructed as follows:

The “main” operator is the + sign between the two parentheses and goes at the root, splitting the expression into left and right subtrees.

The left subtree is constructed from (7 + 10/5). The “main” operator is again the + sign, and the 7 goes left, 10/5 goes right. Now the “main” operator is /, and that is the root of the subtree. Similarly going right at the root.

# Exercises

1. (a) ((a \* b) – (c – d) + e) / (f \* g)

(Check:)

1 3 2 7 4 6 5 9 8 13 10 12 11

(( a \* b ) - ( c - d ) + e) / ( f \* g )

(b) Because the operands and operators are then held in the **correct order** to be **evaluated** using a **stack**.

2. (a) (i) (a +b) \* c / (d + e)

(ii) a b + c d e + / \*

(b) infix and post-fix notation

Chapter 56 - Structure of the Internet

# In-text questions

**Q1:** Answers include:file transfer via ftp, email using a non-web-based email client, remote access and control of a computer / telnet services, usenet groups, smartphone apps, peer-to-peer file sharing networks and instant messaging.

**Q2:** Domain names are much easier for humans to remember and type in than a string of numbers in an IP address.

# Exercises

1. (a) The host of the requested resource [1]  
(b) Domain name. Reject FDQN since the host is not given in this section [1]  
(c) the path or location of the file/resources or description of the folder structure [1]

2. (a) Provide a marketplace for available domain names, manages available domain names, dispute resolution services (cybersquatting), registers individual or company details such as name, address, date of registration or type of organisation. [3]

(b) An ISP sells clients Internet access / connection to the Internet backbone [1]

Reject: Provides an Internet service

Chapter 57 - Packet switching and routers

# In-text questions

**Q1:** The IP address of the sender is included in order to identify the sender, to send a request back to the sender for a duplicate copy of the packet if it contains data transmission errors or to notify the sender if the packet cannot be delivered. Firewalls also use the sender’s IP address in their packet filtering.

**Q2:** Data packet headers contain details of the number of packets in the transmission and the sequence number, e.g. package 1 of 8.

# Exercises

1. (a) Message is split into packets / chunks

Each packet is given destination and source IP addresses

Each packet is dispatched to the Internet through a router or gateway

Packets are sent independently

Packets are given a sequence number

Routers forward packets to the destination

Path of transfer determined by routers

Packets reassembled at destination [2]

(b) Data packet headers contain: the sender's IP address, the recipient’s IP address, the number of packets the message has been broken into, the identification number of the particular packet, the protocol being used, the packet length (on networks that have variable length packets) and the time to live (the number of links or hops that the packet can be routed before being allowed to expire). [3]

(c) A router operates between two networks using the same protocols.  
 A gateway connects two networks using different protocols. [2]

Chapter 58

Internet security

# In-text questions

**Q1:** If either party loses the key or it is stolen, the system is broken. If the key is also intercepted along with the data, the message can be deciphered.

**Q2:** Answers for the reasons for include:

Detection of illegal activities, protection of national interests by monitoring electronic messages

Reasons against include:

Invasion of personal privacy and the desire for companies to maintain company secrets.

**Q3:** Theoretically, a programmer would not sign a program containing any malicious code, so all signed files should be ‘safe’. A ‘signed’ message is very difficult to deny having sent.

**Q4:** ILOVEYOU was a Visual Basic script worm created in 2000 that attached itself to an email sent to all of your address book contents with the title ‘ILOVEYOU’ with an attachment entitled ‘LOVE-LETTER-FOR-YOU.txt.vbs’. Curiosity would have caused many of your friends to open this, activating the virus and subsequently sending copies to each of their address book contacts and so on.

Melissa was a macro virus in 1999 that operated on similar principles, copying itself to the first 50 contacts in your address book when the host file was opened and disabling several safeguards in Office applications.

Blaster was a worm that spread itself through a vulnerability in Windows memory buffers in 2003. It caused a denial of service (DOS) attack on Microsoft’s update site.

Cascade was a memory resident file infector virus prevalent in the 1980s and 90s that caused characters in screen text to fall from top to bottom.

See: http://wildammo.com/2010/10/12/10-most-destructive-computer-worms-and-viruses-ever for more information.

# Exercises

1. (a) (Using an algorithm) to convert a message into a form that is not understandable

(without the key to decrypt it);

(Using an algorithm) to convert a message into a form that is only understandable by the intended parties // can only be read with the correct key;

Converting a message into cipher text;

**NE** scrambling unless further explanation is provided

**A** “unreadable” for “understandable”

**A** “data” for “a message”

**MAX 1**

(b) (i) B will not be able to decrypt it // A‟s private key would be needed to decrypt it // only A could decrypt it; (as ...)

Only A has access to A’s private key // B cannot access A’s private key;

**MAX 1**

(b) (ii) As A’s public key is available to anyone;

Anybody could decrypt it;

**MAX 1**

(c) **Subject-related points:**

**Purpose:**

To authenticate/confirm identity of sender // that message was sent by A // To detect if message has been tampered with/changed;

**How used:**

\*1 Hash/digest produced/calculated from message // (shortened) value calculated from message; **A** message is hashed **A** message digest created

\*1 Hash encrypted with A’s private key;

\*1 Encrypted hash is known as the (digital) signature;

\*2 (Digital) signature is appended to message; **A** transmitted with message **A** even if stated or implied that this is done after the encryption of the message using B’s public key **A** hash or digest

A encrypts message and signature with B’s public key; A without reference to signature but **TO** if clear from order of statements or what candidate has written that the signature is not encrypted with B’s public key

B decrypts message and signature with B’s private key; **A** without reference to signature

B decrypts (digital) signature using A’s public key (to reveal hash);

B reproduces/recalculates hash from received message; **A** re-hashed **A** creates new digest

\*3 If received hash matches reproduced hash then message has not been tampered with // identity of sender is authenticated;

**A** Data for message

**A** Digest, checksum for hash

**A** Encrypted hash/Encrypted digest for signature

**A** Example of hashing method e.g. MD2/4/5/6, SH0/1/224/256/384/512

\*1 = as an alternative to these three points, allow one mark for the idea that the digital signature is calculated from/hashed from/a digest of the message

\*2 = only award this mark if there is previously the concept of the hash or signature being produced.

\*3 = can only be awarded if there is clear concept that the comparison is to a recalculated hash

**Only one mark should be awarded for the purpose. Other marks must come from how the digital signature is used.**

The purpose mark could be implicit in the how used mark and should be awarded if it is.

It is acceptable for steps to be missed out.

Accept responses with message sent from B to A if it is clear that this is what the candidate has done.

|  |  |
| --- | --- |
| **How to award marks: Mark Bands and Description** | |
| 5-6 | *To achieve a mark in this band, candidates must meet the subject criterion (SUB) and all 5 of the quality of language criteria (QWCx).*  *SUB* Candidate has covered both the purpose and the use of digital signatures, and has made at least five subject-related points including both creation and use.  **To get 6 marks, the answer must include reference to the encryption of the message digest/hash using A’s private key.**  *QWC1* Text is legible.  *QWC2* There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear.  *QWC3* The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently.  *QWC4* Sentences (and paragraphs) follow on from one another clearly and coherently.  *QWC5* Appropriate specialist vocabulary has been used. |
| 3-4 | *To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of language criteria (QWCx).*  *SUB* Candidate has provided a description of some parts of the process and has made at least three subject-related points.  *QWC1* Text is legible.  *QWC2* There may be occasional errors of spelling, punctuation and grammar. Meaning is clear.  *QWC3* The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently.  *QWC4* The candidate has used well-linked sentences (and paragraphs).  *QWC5* Appropriate specialist vocabulary has been used. |
| 1-2 | *To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of language criteria (QWCx).*  *SUB* Only one or two relevant points have been made.  *QWC1* Most of the text is legible.  *QWC2* There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response.  *QWC3* The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed.  *QWC4* Sentences (and paragraphs) may not always be well- connected.  *QWC5* Specialist vocabulary has been used inappropriately or not at all. |
| 0 | Candidate has made no relevant points. |

Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question.

If a candidate meets the subject criterion in a band but does not meet the quality of language criteria then drop mark by one band, providing that at least 4 of the quality of language criteria are met in the lower band. If 4 criteria are not met then drop by two bands.

2. (a) Filtering, firewalls, use of cached ‘safe’ sites

(b) Stateful inspection // dynamic filtering examines the payload contents of each packet; creates contextual //temporary rules to keep open only the ports that are currently being used;

(c) Answers include: regular updates to malware protection software // antivirus software; updates // patches to operating system software; strict email filtering; website filtering // firewall filtering; training in handing email attachments // downloading files // file transfer, **NE** acceptable use policies, must elaborate

Chapter 59 – TCP/IP

# In-text questions

**Q1:** The consignment represents data being sent; using the ‘shipping agent protocol’ (application layer); this is split into five packets of 1000 widgets or Kb. The name of your friend (the port number) is given on a cover note inside the boxes, and ‘Box n of 5’ is written on the sides (transport layer). The shipping agent prints the ‘To’ and ‘From’ street addresses (IP addresses) on another label and adds this to the boxes (network layer). Lastly the boxes are forwarded using a delivery driver (link layer) to an intermediary depot (a router, back on the network layer) using a depot ID number (MAC address) for overnight storage.

The depot then replaces the depot ID with the destination ID and forwards the consignment on again using another driver (link layer) to the destination. The agents labels are removed from the boxes upon delivery, the boxes are inspected and rearranged into the correct order. If a box is damaged or missing, the sender is informed and a replacement box is sent. The boxes are then opened and the widgets (data) are assembled into a whole part.

**Q2:** The port address is 443. A protocol is an agreed set of rules. HTTPS is the protocol that uses this port, and would commonly be used for secure online transactions such as shopping or banking.

**Q3:** Telnet is a protocol that allows remote access to a computer, commonly for administrative management. SSH has superseded Telnet as a modern equivalent since it uses encryption, which Telnet does not.

**Q4:** Georgina uses SMTP to send her message; this goes via her local email server; which checks the domain to which the email is addressed e.g.@domain.com; and forwards the message (often via other mail servers); to Nick’s webmail server. Nick logs in via his local ISP; and downloads his message using POP3/IMAP.

# Exercises

1. Concept that data passed up/down between layers; A by example – just one needed but must be correct **NE** just describing the layers in the correct order

Application layer selects appropriate protocol for the communication / protocol mentioned by example;

The role of the application layer is to interact with the user via appropriate application software (eg web browser / ftp client) or the users system (eg synchronising files);

Transport layer establishes end to end communication // Transport layer establishes a virtual path;

Transport layer deals with error control (acknowledgements\retransmission) / segmentation / flow control

Communication split into packets by transport layer // re-assembled by receiver;

Packets are numbered by transport layer;

Transport chooses a Port number for client and destination;

Network/IP layer supplies appropriate IP addresses for source and destination (when sending packets);

Network/IP layer involved with packet routing / moving datagrams to the next network node (router);

Combination of IP address and Port = Socket / described;

Link layer receives packets from network layer and adds MAC addresses; A hardware address for BOD

Link layer moves packets between 2 internet hosts;

Link layer adds frame header and footer to packets;

Link layer deals with physical connection/cabling;

**A** Link layer includes network card / drivers;

Network/IP layer strips IP addresses (if receiving packets) // Link layer strips MAC address (if receiving);

Idea of encapsulation described re datagram;

[8]

|  |  |  |
| --- | --- | --- |
| **Mark Bands and Description** | |  |
| 6-8 | To achieve a mark in this band, candidates must meet the subject criterion (SUB) and all of the quality of language criteria (QLx).  SUB Candidate has provided a clear explanation of principles of operation, including at least 6 of the points listed above and at least 3 distinct levels of the TCP/IP stack.  QL1 Text is legible.  QL2 There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear.  QL3 The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently.  QL4 Sentences and paragraphs follow on from one another clearly and coherently.  QL5 Appropriate specialist vocabulary has been used. |  |
| 3-5 | To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of language criteria (QLx).  SUB Candidate has provided a limited explanation of principles of operation, including at least 3 of the points listed above and at least two distinct levels of the TCP/IP stack.  QL1 Text is legible.  QL2 There may be occasional errors of spelling, punctuation and grammar. Meaning is clear.  QL3 The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently.  QL4 The candidate has used well-linked sentences and paragraphs.  QL5 Appropriate specialist vocabulary has been used. |  |
| 1-2 | To achieve a mark in this band, candidates must meet the subject criterion (SUB). The quality of language should be typified by the QLx statements.  SUB Candidate has provided a weak explanation which covers at least 1 of the points listed above for 1 mark or 2 points to get 2 marks and at least one distinct level of the TCP/IP stack.  QL1 Most of the text is legible.  QL2 There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response.  QL3 The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed.  QL4 Sentences and paragraphs may not always be well-connected or bullet points may have been used.  QL5 Specialist vocabulary has been used inappropriately or not at all. |
| 0 | Candidate has not made reference to any of the points listed above. |

Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question.

**IF A CANDIDATE MEETS THE SUBJECT CRITERION IN A BAND BUT DOES NOT MEET THE QUALITY OF LANGUAGE CRITERIA THEN DROP MARK BY ONE BAND, PROVIDING THAT AT LEAST 3 OF THE QUALITY OF LANGUAGE CRITERIA ARE MET IN THE LOWER BAND. IF 3 CRITERIA ARE NOT MET THEN DROP BY TWO BANDS.**

2. a) i) To manage/control/execute commands on a remote machine;

A remote access/login

A – a clear example of remote management

NE remote viewing

R remote desktop

1. Enable files on one host/computer/client to be copied to another host/computer/server;

To manage files on a remote computer/server;   
A to upload/download/transfer files   
NE “sharing”   
NE load a file   
NE transfer data

iii) To retrieve/fetch (stored) email;  
To check for new emails;   
A access/download/receive   
R sending TO any mention of sending   
NE just “email”

b) i) 192.168.3.205 // 74.125.4.148 // 208.43.202.29; [1]

ii) 80 // 25 // 58539 // 57458 // 57459; [1]

iii) 192.168.3.205:80 //  
192.168.3.205:25 //   
74.125.4.148:58539 //   
208.43.202.29:57458 //   
208.43.202.29:57459 ; [1]

c) Servers might be in another room / site / cupboard / inaccessible ; Servers might not have a keyboard / monitor installed ;   
Can manage multiple servers from one machine;   
Servers can be managed outside of work hours / from anywhere;   
It would be quicker (A more convenient) (to manage from her machine than visit the servers) // better time management;   
Server rooms are often uncomfortable places for people to work in;   
NE she does not need to go to the servers

Chapter 60 – IP addresses

# In-text questions

**Q1:** In a classful system, Class C would offer a smaller range of host IDs which would likely be more suitable for a small business with fewer than 254 networked devices. (Note: 256 unique host addresses but 00000000 and 11111111 are unusable, so subtract two from 256 = 254.)

**A classless syetm would allow greater flexibility.** 25 users are likely to have a range of other addressable devices within the network such as printers. For this reason at least 32 addresses would be needed, if not 64. www.xxx.yyy.zzz/26 leaving 6 bits (26) for 64 host addresses for example.

**Q2:** Any IP address beginning with 172.16.10.x where x is anything between 1 and 254. Do not allow 0 and 255 as these are reserved for the generic network segment address and the broadcast address.

**Q3:** A combined home router/hub device requires a public IP address to communicate with other devices on the Internet and to be routed to. It requires a private IP address (often 192.168.0.1) to connect to the same internal subnet as other computers on your home LAN.

# Exercises

1. (a) (i) 192.168.0.x where x is not 0 or 255. **R** addresses that include ports numbers.

(ii) 192.168.2.x where x is not 0 or 255. **R** addresses that include ports numbers.

(iii) 192.168.2.y where y is not 0 or 255 and not the same as x in part (ii).

(iv) Answers in the dotted-decimal format w.x.y.z where the following rules apply:

w, x, y and z are between 0 and 255 inclusive,

w is not 0 or 127,

z is not 0 or 255,

(b) Public addresses are globally unique // private addresses are not unique.

Public addresses can be connected to directly from an external network such as the Internet // private addresses require a gateway or router to connect through.

Public addresses are allocated by a central Internet registry // private addresses are allocated by a network administrator

(c) (i) Reduces network data collisions // on a bus network because two computers in one segment can communicate at the same time as two computers on another segment. Or because the broadcast domain is reduced.

Increases network security; by localising packet transmission to one segment

Smaller segments of the network // A subnetwork for each department within the organisation are more reliable and simpler to administer/troubleshoot/manage; cable/device failure // performance issues are limited to one segment.

(ii) 32-bit IP address – 24 bits – 2 bits = 6 bits. 62 = 64; Less 2 for all-zero and all-one host addresses (e.g. x.x.x.yy000000 and x.x.x.yy111111) = 62.

(d) (i) IP configuration information is automatically setup for the client machine by the DHCP server // A mobile device/computer moving from one subnet to another will be automatically rediscovered by the DHCP server as soon as it broadcasts a new ‘DHCPDiscover’ request. // Less likelihood of an IP address conflict.

(ii) IPv4 addresses are running out; dynamic addressing shares the remaining available IP addresses between active users.

(e) Port forwarding is required because: an FTP server operating in a public address space; cannot directly access a host on a private network;

The NAT forwards inbound data packets using a particular IP address and port (socket address) back to a specific port (allow examples port 20/21) that the private host within the LAN is using.

Chapter 61 – Client server model

# In-text questions

**Q1:** An online auction site would benefit from the WebSocket protocol because:

Bidders could maintain a persistent link throughout the duration of the auction without fear of disconnection;

Bid data will travel between the client and the server much faster owing to smaller packet overheads offering a real-time experience;

Less chance of two bidders ‘winning’ the item with real-time data;

Much less pressure on website servers since the header data size can be reduced by up to 1000-1.

**Q2:** Create (POST) a new contact to the address book;

Retrieve (GET) the details for an existing contact;

Update (PUT) *e.g.* the new telephone number for contact 1 in the record;

Delete (DELETE) a contact from the database.

**Q3:** POST http://solarsystem.com/planets/Venus

**Q4:** {"planets":[

{"name": "Mars",

"distanceFromSunKM": 227900000,

"gravityMS2": 3.71},

]}

Allow answers with each field and value on a single line.

**Q5:** A navigation app with all of the processing and search and storage operating locally on the device will not suffer problems with a frequent loss of connection whilst on the move. It will also require constant access to the device’s own hardware such as a compass and GPS which are easier to use if the app lives on the device in a thick-client setup. The use of mobile data is also likely to be less with a thick-client system and therefore the load on servers running the system is likely to be much less.

However, a thin-client system may offer less powerful phones the opportunity to run high-powered apps.

Exercises

1. (a) GET /animal/aardvark (Award one mark for GET and another for including

animal/aardvark at the end of a sensible URL.)

(b) Response A is written in JSON format.

(c) Easier for humans to read;

More compact;

Easier to create;

Easier for computers to parse // quicker to parse // JS or Python function can process it directly.

2. (a) Client computer sends a request to a server;

The server receives the request and responds with the requested data;

(b) Client terminals load operating system from server at start up;

Applications installed on server and accessed remotely by terminals;

Client (dumb) terminals act as interfaces only;

All processing is done by the server;

(c) Server must have multiple processors // lots of RAM;

Basic processor // no storage device required in client terminals;

Higher bandwidth network connections required to cope with demand on server;

Network must use a switch not a hub;

3. Full duplex // bi-directional // two-way connection is established;

Persistent connection until the communication is terminated;

Reduced header information, increasing data transfer speed; and reducing load on server;

Uses may include: (*Application; Justification.*)

Social feeds; Instant information updates in real-time;

Multiplayer online games; high data traffic; efficient and immediate interaction between players is crucial;

Collaborative cloud-based editing/coding; real-time view of who is editing what // instant updates;

Clickstream data (web statistics); ability to track and playback customer mouse moves;

Financial tickers // share data; live data viewing for trading information;

Sports updates; real-time scores without latency;

Multimedia chat // videoconferencing; no latency; reduced data load = faster transfer;

Location-based apps; instant location updates for tracking;

Online education; live interaction; collaborative notes // chat etc.

Chapter 62 – Entity relationship modelling

# In-text questions

**Q1**: Some suggestions: Date of birth, gender, Date next check-up due, date and time of next appointment, DentistID, NI Number, British resident(Y/N), Nationality, private/NHS, next of kin, allergies, medical conditions

**Q2:** The entityAppointmentmay have attributes Date, time, length of appointment, PatientID, DentistID, treatment code

**Q3:** NI number is not a suitable primary key because many patients may not know their NI number and some patients may not have an NI number, e.g. if not British.

**Q4:** Many different nurses treat many different patients but this is probably not a relationship that needs to be recorded.

**Q5:** (a)

has

Component

Product

(b)

Product

Order

Customer

has

specifies

An extra entity, OrderLine, will be needed to implement the many-to-many relationship between Order and Product. Similarly, a third table will be needed to link Product and Component in a relational database.

# Exercises

1. (a) Any three reasonable answers, such as ViewingID, Date, Time, PropertyID, BuyerID, staffID [3]

(b) Property (PropertyID, address, propertyType, bedrooms, askingPrice, *VendorID*) [2]

Vendor (VendorID, name, address, telephone) [1]

Buyer( BuyerID, name, address, telephone, PropertyTypeRequired, LowerLimit, UpperLimit) [2]

Viewing (ViewingID, Date, Time, *PropertyID*, *BuyerID*, *staffID*) [3]

(Must show all primary keys and foreign keys for full marks)

(c)

Property

Viewing

has

Vendor

Buyer

has

owns

2 (a)

Member

Loan

Book

has

is for

(b) Composite key

(c) A foreign key is an **attribute** which is a **primary key in another table**.

MemberID, BookID are both foreign keys.

3. (a) ExamPaperID [3]

(b) CandidateCourse(*candidateID, courseID*)

(c) [5]

Candidate

CandidateCourse

Course

Results

ExamPaper

(d) Results (CandidateID, *ExamPaperID*, *CourseID,* MarkReceived)

Chapter 63 – Relational databases and normalisation

# In-text questions

**Q1:**  It is not a good idea because it wastes space if products have fewer than four components, and if they have more than four, they cannot all be held in the table.

**Q2:**

**Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Product ID** | **Product Name** | **CostPrice** | **Selling Price** |
| 123 | Small Monkey | 2.50 | 5.95 |
| 156 | Pink Kitten | 3.10 | 6.00 |

**ProductComp**

|  |  |  |
| --- | --- | --- |
| **Product ID** | **Comp ID** | **Comp Qty** |
| 123 | ST01 | 30 |
| 123 | G56 | 2 |
| 123 | FF77 | 0.3 |
| 156 | ST01 | 45 |
| 156 | G120 | 2 |
| 156 | FF88 | 0.35 |
| 156 | S34 | 1 |

**Component**

|  |  |  |  |
| --- | --- | --- | --- |
| **Comp ID** | **CompName** | **Supplier ID** | **Supplier Name** |
| ST01 | Stuffing | ABC | ABC Ltd |
| G56 | Eye (small) | BH Glass | Brown & Hill |
| FF77 | Brown Fur | FineFur | Fine Toys Ltd |
| G120 | Eye (medium) | XYZ Glass | XYZ Ltd |
| FF88 | Pink fur | FineFur | Fine Toys Ltd |
| S34 | Soundbox | Ping Toys | Ping & Co |

**Q3:** ProductID, CompID in ProductComp table

**Q4:** Event (EventID, EventName)

EventInYear (*EventID*, Year, Winner, TimeOrDistance)

Event

EventInYear

# Exercises

1. (a) CatalogueNo

(b) To be in first normal form the table must contain no repeating attribute or groups of attributes. This table contains repeating groups of attributes as each CD has several tracks and track attributes, songs with song attributes and artists with artist attributes

(c) [4]

CompactDisc

SongMusic

Artist

CD-Track

(d) CompactDisc (CatalogueNo, Title, RecordCompany, Type) [2]

SongMusic (SongMusicID, SongMusicTitle, Composer) [2]

Artist (ArtistID, ArtistName) [2]

CD-Track (CatalogueNumber, TrackNo, TrackDuration, SongMusicID, ArtistID) [4]

(e) SELECT CatalogueNumber, TrackNo, SongMusicID, ArtistID

FROM CD-Track

WHERE (CatalogueNo = “15438”) [3]

2. (a) **Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **StudentNumber** | **StudentName** | **DateOfBirth** | **Gender** |
| 1111 | Bell, K | 14-01-1998 | M |
| 2222 | Cope, F | 12-08-1997 | F |
| 3333 | Behr,K | 31-07-1996 | M |

**Course**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Number** | **CourseName** | **TeacherID** | **Teacher Name** |
| COMP23 | Java1 | 8563 | Davey,A |
| G101 | Animation | 1567 | Day,S |
| Comp16 | Intro to OOP | 2299 | Ross,M |
| Comp34 | Database Design | 3370 | Blaine, N |

**StudentCourse**

|  |  |
| --- | --- |
| **Student Number** | **Course Number** |
| 1111 | COMP23 |
| 2222 | COMP23 |
| 2222 | COMP16 |
| 2222 | G101 |
| 3333 | Comp16 |
| 3333 | Comp34 |

[6]

(b) A fully normalised database must have no repeating attributes or groups of attributes

It must contain no non-key dependencies or partial key dependencies. [2]

Chapter 64 – Introduction to SQL

# In-text questions

**Q1:** SELECT \*

FROM CD

WHERE ((RecordCompany = “ABC”) OR (RecordCompany = “GHK”))

AND (DatePublished BETWEEN #01/01/2014# AND #31/12/2015#)

The CD numbers of the records returned are:

CD14356, CD19998, CD25364 and CD 77233

**Q2:** SELECT Song.SongTitle, Artist.ArtistName, Song.MusicType

FROM Song, Artist

WHERE Song.ArtistID = Artist.ArtistID

AND (Artist.ArtistName = "JJ" OR Artist.ArtistName = "Fred Bates")

ORDER BY Artist.ArtistName, Song.SongTitle

This will produce the following results:

|  |  |  |
| --- | --- | --- |
| **SongTitle** | **ArtistName** | **MusicType** |
| Ghost Town | Fred Bates | Heavy Metal |
| Gentle Waves | Fred Bates | Heavy Metal |
| Complicated Game | JJ | Americana |
| Waterfall | JJ | Americana |

Exercises

1. (a) [3]

Customer

Order

Product

OrderLine

(b) (i) Product (ProductID. Description, QtyInStock) [2]

(ii) Customer (CustomerID, Name, Address, Telephone) [2]

(iii) Order (ABCOrderNo, CustomerOrderNo, CustomerID, Despatched) [3]

(iv) OrderLine (ABCOrderNo, OrderLine, ProductID, QtyOrdered) [4]

(c) SELECT Customer.CustomerName

WHERE (Customer.CustomerID = Order.CustomerID)

AND (Order.Despatched = TRUE)

ORDER BY Order.ABCOrderNo [6]

2. (a) [4]

Trip

PupilTrip

Pupil

Teacher

(b) (i) SELECT Trip.Description, Trip.StartDate, Trip.TripID, Pupil.PupilID, Pupil.PupilSurname, Pupil.PupilFirstName

FROM Pupil, tblPupilTrip, Trip

WHERE (Pupil.PupilID = PupilTrip.PupilID) AND (Trip.TripID = PupilTrip.TripID) AND (Trip.TripID="14"); [4]

(ii) SELECT Teacher.TeacherID, Teacher.Title, Teacher.Surname, Trip.Description, Trip.StartDate

FROM Teacher, Trip

WHERE Teacher.TeacherID = Trip.TeacherInCharge AND Teacher.Surname="Black"

ORDER BY Trip.StartDate Desc; [4]

(iii) SELECT Pupil.PupilID, Pupil.PupilSurname, Pupil.PupilFirstName, Trip.TripID, Trip.Description, Trip.StartDate, Teacher.FirstName, Teacher.Surname

FROM Teacher, Pupil, PupilTrip, Trip

WHERE Teacher.TeacherID = Trip.TeacherID AND

Pupil.PupilID = PupilTrip.PupilID AND Trip.TripID = PupilTrip.TripID AND (Trip.Description Like "\*Year 7\*") AND   
(Trip.StartDate Between #5/1/2015# And #5/30/2015#); [6]

Chapter 65 – Defining and updating tables using SQL

# In-text questions

**Q1:**  CREATE TABLE Student

(

StudentID VARCHAR(6) NOT NULL,

Surname VARCHAR(20),

FirstName VARCHAR(15),

DateOfBirth DATE,

PRIMARY KEY StudentID

)

**Q2:** ALTER TABLE Student

ADD YearGroup INT

**Q3:** CREATE TABLE Course

(

CourseID CHARACTER (6) NOT NULL,

CourseTitle VARCHAR(30) NOT NULL

CourseDate DATE,

PRIMARY KEY CourseID

)

Q4: (a) INSERT INTO Student (StudentID, Surname, FirstName, DateOfBirth)

VALUES (‘AB1234’,’Daley’,’Jennifer’,#23/06/2005#)

(b) UPDATE Student

SET FirstName = ‘Jane’

(c) ALTER TABLE Student

ADD DateStarted DATE

Q5: The record will have the original credit limit and the updated address, as the updated credit limit will be overwritten by User A’s update.

# Exercises

1. (a) (i) What it means:

Every attribute in the relation is dependent on the key; the whole key and nothing but the key.

OR relations contain no repeating groups of attributes

There are no partial dependencies

No non-key dependencies

(ii) Why is it important:

Eliminate update anomalies (or give an example)

Eliminate insertion anomalies (or give an example)

Eliminate deletion anomalies (or give an example)

Eliminate data inconsistency

Minimise/reduce data duplication

**MAX 2**

(b)

Furniture

CustomerOrder

Customer

CustomerOrderLine

[3]

MAX 2 if any incorrect relationships drawn; MAX 3

(c) *There are several variations on the solution – see “Marking” below*

CREATE TABLE Furniture

(

FurnitureID INT,

FurnitureName VARCHAR(30),

Category VARCHAR(10),

Price SMALLMONEY,

SupplierName VARCHAR(20),

PRIMARY KEY FurnitureID

)

*Marking:*

FurnitureID INT PRIMARY KEY NOT NULL (note that “NOT NULL” is optional)

or

FURNITUREID INT

PRIMARY KEY (FurnitureID)

FurnitureName VARCHAR(30)

CATEGORY VARCHAR(10)

Price SMALLMONEY (note that currency is not a valid SQL type.)

SupplierName VARCHAR(20)

1 mark for FurnitureID, with sensible data type and identified as primary key

1 mark for two other fields with sensible data types and lengths OR 2 marks for all four other fields with sensible data types and lengths

Valid alternative types for FurnitureID are smallint, mediumint, integer, any text type

Alternative types for Price are money, float, real, decimal, double, numeric, int, smallint, mediumint, integer

Alternative types for text fields are char, varchar, nchar, nvarchar, text etc.

Ignore punctuation errors. [3]

(d) SELECT CustomerName, TelephoneNumber

FROM Customer, CustomerOrder, CustomerOrderline

WHERE FurnitureID=10765

AND Customer.CustomerID= CustomerOrder.CustomerID

AND CustomerOrder.OrderID= CustomerOrderLine.OrderID

ORDER BY CustomerName (ASC)

1 mark for correct two fields in SELECT clause

1 mark for correct three tables in FROM clause

1 mark for FurnitureID = 10765

1 mark for Customer.CustomerID = CustomerOrder.CustomerID, joined to other conditions with AND

1 mark for CustomerOrder.OrderID = CustomerOrderLine.OrderID, joined to other conditions with AND

1 mark for ORDER BY CustomerName, ASC is optional

--- OR ---

SELECT CustomerName, TelephoneNumber

FROM Customer INNER JOIN CustomerOrder

ON Customer.CustomerID=CustomerOrder.CustomerID INNER JOIN CustomerOrderLine

ON CustomerOrder.OrderID=CustomerOrderLine.

OrderID WHERE FurnitureID = 10765 ORDER BY CustomerName (ASC)

1 mark for correct two fields in SELECT clause 1 mark for correct three tables in FROM clause

1 mark for INNER JOIN using Customer.CustomerID=CustomerOrder.CustomerID 1 mark for INNER JOIN using CustomerOrder.OrderID=CustomerOrderLine. OrderID

1 mark for FurnitureID = 10765

1 mark for ORDER BY CustomerName, ASC is optional

Marks for SELECT and FROM statements should not be awarded if additional fields/tables included. Marks can be awarded for the conditions in the WHERE statement even if the required tables are not present in the FROM. Accept FurnitureID with no quotation marks, single quotation marks or double quotation marks.

Accept table names before fieldnames. Accept use of Alias/AS command

e.g. FROM Customer AS C then use of C as table name.

Accept insertion of spaces into fieldnames Ignore unnecessary clause CustomerOrderLine.FurnitureID=Furniture.FurnitureID I unnecessary brackets

DPT for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause.

DPT for fieldname before table name. [6]

2. (a) User 1 access a record, which is then copied to the client computer where they update it. Before they save, User 2 accesses the same record from the server, updates it on their computer and saves it. User 1 then saves their updated record, which overwrites User 2’s update. [3]

(b) Deadlock is when User 1 accesses Record A which is then locked. User 2 then accesses Record B which is locked. They then each attempt to access the locked records before releasing the records they have opened, so that neither can proceed. [2]

(c) Serialisation, time stamp ordering, commitment ordering. Each database object has a timestamp to say when it was last read or written. The DBMS ensures that records are correctly updated in the right order, or one of the transactions is aborted. [2]

Chapter 66 – Systematic approach to problem solving

# In-text questions

**Q1:** Draw a family tree.

# Exercises

1. (a) The prototyping agile approach involves a high degree of user interaction and feedback (1), at every stage of design and implementation (1), with incremental changes being made (1) to ensure the system delivered matches user requirements (1) which may not be fully comprehended by either user or developer (1) at the start of the project. It may involve building a prototype user interface (1).

(b) It ensures that errors and misunderstandings are corrected at the earliest possible stage, (1) it helps both user and developer to understand exactly what is required and what can be delivered, (1) it allows for changes to the specification when problems/omissions arise, (1) and results in a system that the user is likely to be satisfied with (1).

Any other reasonable points.

2. (a) Interviewing potential users, observing current systems, looking at existing reports etc. if there are any.

Building a prototype system, or part of the system, and showing it to the user to highlight any misunderstandings or weaknesses. Reworking in response to user feedback.

(b) The designer needs to find out what entities, attributes and relationships are involved and draw diagrams of how all the entities will relate to each other. Sample data can be used to clarify what data items need to be held and the relationships between entities.

Chapter 67– Basic Concepts of object-oriented programming

# In-text questions

**Q1**: (a) Cat: At a vet’s for example: Name, date of birth, breed, sex, weight, state of health, name and address of owner, vaccinations etc.

(b) Rectangle: length, breadth, line thickness, line colour, fill colour, position

(d) Hotel booking: Date, NoOfNights, RoomNo/RoomType, BreakfastRequired, Name, etc.

**Q2:** Class definition and constructor for Radio:

\* Radio class definition

Radio = Class \* This is the class header

\* instance variables

Private

Volume: Integer

Station: String

Switch: Boolean

\* method headers

Public

Procedure SetVolume(Integer aVolume)

Procedure SetStation(String aStation)

Procedure SetSwitchOn

Procedure SetSwitchOff

Function GetVolume

Function GetStation

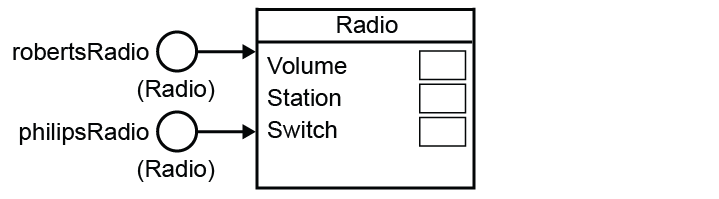
Function GetSwitch

To instantiate two new objects:

robertsRadio = new Radio

philipsRadio = new Radio

**Q3**: Variable reference diagram:



**Q4**: philipsRadio.SetSwitchOn, robertsRadio.SetStation(“BBC2”)

(or a similar answer, depending on how the method for SetSwitchOn has been defined)

**Q5**: makeNest, hide, squeak, eatCheese etc.

**Q6**: Subclass definition for Beaver:

Animal = Class

Public

Procedure MoveLeft(Integer Steps)

Procedure MoveRight (Integer Steps)

Procedure GetColour(String Colour)

Function GetPosition

Function GetColour

Private

Position: Integer

Colour : String

End

Rodent = Subclass(Animal)

Public

Procedure Gnaw

End

Beaver = Subclass (Rodent)

Public

Procedure MakeDam

Procedure CutTree

Function GetTreesCut

Function GetDamComplete

Private

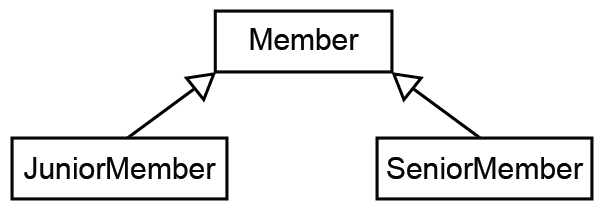
TreesCut: Integer

DamComplete: Boolean

End

# Exercises

1. (a)



(b) Member = Class

Public

Procedure AddNewMember(Integer anID, String aFirstName, String aSurname, String aTelephone)

Procedure DeleteMember

Function GetMemberDetails

Private

MemberID: Integer

FirstName: String

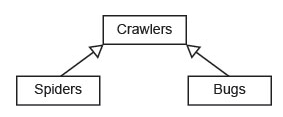
Surname: String

Telephone: String

End

(c) Encapsulation means that both the data and the methods are stored with the object.

2 (a)



(b) (i) Attributes: Colour, Position, Type, Gender, Size, etc.

(ii) CatchFly, SpinWeb, etc.

Chapter 68 – Object-oriented design principles

# In-text questions

**Q1**: (a) Aggregation – the animals will (hopefully) be rehoused when a zoo closes down.

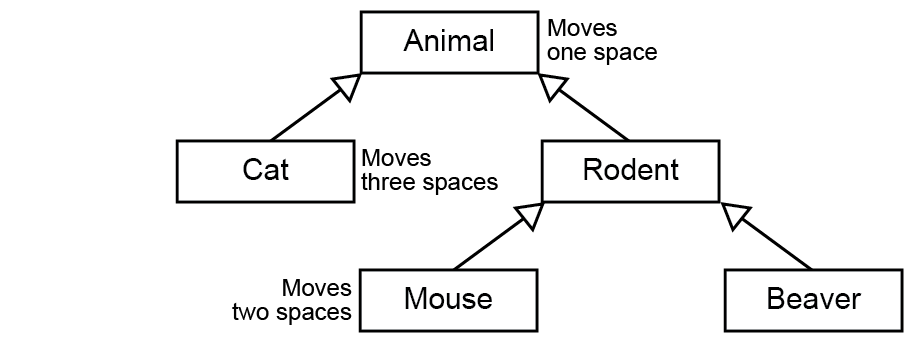
(b) Composition – the track sections will cease to exist if for example the race track is built over.

(c) Aggregation – teachers will be moved to another department (or school) if their department closes down

**Q2:** tom will move right 3 spaces

jerry will move right 2 spaces

**Q3:** The Mouse class, not the Rodent class, needs to override the moveRight and moveLeft methods defined in the Animal class. In the Mouse class, these methods will cause a move of 2 spaces.



# Exercises

1. (a) A class/subclass has/shares/inherits properties and methods with the (parent) class (it is derived from); **A** another class

Building a hierarchy of classes with each child class inheriting access to its parent class's methods and properties;

Relationship between two object types/objects in which one object (type) is a kind of the other;

**MAX 1**

**A** Just one of properties and methods, do not need both.

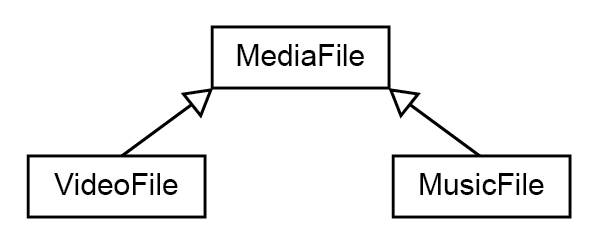
**A** The following as alternatives to properties: fields, attributes, characteristics, data with data as BOD

**A** The following as alternatives to methods: procedures, functions, code.

**A** The following as alternatives to parent: base, super.

**A** The following as alternative to child: descendent, subclass, derived. [1]

(b)



**1 mark** for class names in boxes, with MediaFile drawn above the other two;

**1 mark** for correct arrows;

**A** arrows drawn as:

**A** filled/empty arrowheads

**A** rotated through 90 degrees [2]

(c) Method can be defined with same name; **A** method can be redefined, an inherited method (but not just inheritance) as implying same name

But have different implementation/code // perform different function;

The redefined method will be used instead of the parent's method;

**A** This is an example of polymorphism

**A** Procedure, function, subroutine for method. [2]

(d) MusicFile = Class/Subclass (MediaFile) 1

Public

Procedure PlayFile (Override) 1

Function GetArtist

Function GetSampleRate 1

Function GetBitDepth

Private

Artist : String

SampleRate : Real 1

BitDepth : Integer

End

**1 mark** for correct header including name of class and parent class;

**1 mark** for redefining the PlayFile procedure;

**1 mark** for defining all 3 extra functions needed to read variable values;

**1 mark** for defining all 3 extra properties, with appropriate data types in

private section;

**A** any numeric types for SampleRate and BitDepth

**A** answers that indicate separately that each variable is private

**DPT** if any extra functions/procedures/variables included but do not penalise answers that have extra procedures to set variable values.

**DPT** if any of the functions/procedures are private

**I** parameters to methods, minor changes to names that do not affect

clarity, case

**-- OR --**

(Public) class/subclass MusicFile extends/inherits

MediaFile { 1

public void PlayFile (Override) 1

public string GetArtist()

public float GetSampleRate() 1

public int GetBitDepth()

private string Artist

private float SampleRate 1

private int BitDepth

}

**1 mark** for correct header including name of class and parent class;

**1 mark** for redefining the PlayFile procedure;

**1 mark** for defining all 3 extra functions needed to read variable values;

**1 mark** for defining all 3 extra properties, with appropriate data types as

private;

**A** any numeric types for SampleRate and BitDepth

**DPT** if any extra functions/procedures/variables included but do not

penalise answers that have extra procedures to set variable values.

**DPT** if any of the functions/procedures are private

**I** parameters to methods, minor changes to names that do not affect

clarity, case

-- ACCEPT MIXES OF TWO METHODS IF MEANING IS CLEAR –[4]

2. (a) Polymorphism: when classes are inherited, subclasses have the ability to process objects differently but using the same method name. [2]

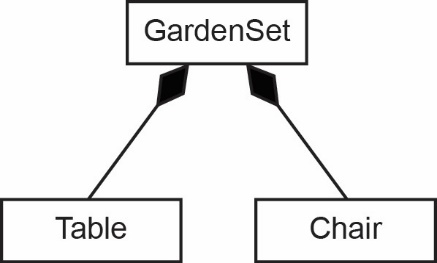
(b) (i) “Birds can fly”

“Seabirds can fly and swim” [2]

(ii) This is an example of polymorphism, where the **move** method in the subclass Seabird overrides the method of the same name in the superclass Bird. [2]

3. (a) Aggregation is a relationship between two classes / a “has a” relationship between two classes, for example, team “has a” player. (Alternative answer: It is a type of association in which a class is a collection or container of other classes which may exist independently of the containing class.)

(b)



(c) The access modifier Private means that only objects in that class will be able to access the variables. Inherited classes will only be able to access instance variables of methods that are declared Public in the superclass.

Making the instance variables Private ensures that other classes can only interact with objects in the superclass via messages. This prevents an inherited class from changing the values of variables used only in the superclass.

Methods are declared Public so that all inherited classes can use them. [3]

Chapter 69 – Functional programming

# In-text questions

**Q1**: Co-domain B is the set of all real numbers greater than or equal 5.

**Q2:** It multiplies the three values. It has three arguments x, y and z. (In the next chapter we will see that every Haskell function officially takes only one parameter, so in this case the single parameter consists of values 2 3 4.)

Applying the function with the arguments 2 3 4 will return the value 24.

**Q3:** sumOfSquares x y = x^2 + y^2

sumOfSquares 3 4

This will return the value 25.

**Q4:** (a) 18

(b) 30

(c) 25

**Q5:** 128

**Q6:** f(a) = 5, g 4 = 64, g.f = g(f(4)) = g(5) = 125

**Q7:** quadruple x = doubleNum (doubleNum x)

**Q8:** 4.0

# Exercises

1. (a) doublePlusOne :: Int -> Int, doublePlusOne x = 2\*x + 1

(b) square x = x\*x *or* square x = x^2

(c) squarexPlusf x = (square x) + (doublePlusOne x)

Value returned = 16

2. (a) (i) “Immutable” means “unchangeable”. A variable in Haskell gets defined only once and cannot change.

(ii) “Stateless” means that the program variables never change state. The program is a sequence of stateless function evaluations – the functions do not change the state of the program.

“No side effects” means that the result of a function is determined only by its input – nothing that happens in the rest of the program has any effect on the result. This means that functions can be evaluated in any order.

(b) Most bugs in imperative programs are caused by side effects – a variable has been changed in some unexpected way by a wrong statement, or perhaps functions are called in the wrong order. If functions in an imperative language use global variables, these may be changed by the function and the programmer may find it hard to trace through to find out why a variable has a certain value. None of this happens in a functional programming language.

Chapter 70 – Function application

# In-text questions

**Q1**: 9

**Q2:** 100 (10 is added to 40 + 50)

# Exercises

1. (a) 2 x 3 x 5 = 30 [1]

(b) (i) 6 x 2 x 10 = 120 [1]

(ii) Partial function application is calling a function with too few parameters, getting back a partially applied function, i.e. a function that takes as many parameters as have been left out.

The function multiply3 is broken down into three functions which each take one argument. Its type declaration is equivalent to

multiply3 :: Integer -> (Integer -> (Integer -> Integer))

multiply3 takes an integer argument and returns a function which multiplies the result by the second argument and returns another function. This function multiplies the result by the third argument and returns the final result. *(2 marks for this explanation)*

multByTen makes use of partial function application, supplying the missing parameter to multiply3. *(1 mark)* Two, instead of three, parameters, are passed to multiply3 *(1 mark)* resulting in a function which multiplies the result 12 by the next parameter. *(1 mark)* The parameter 10 is then applied to the new partially applied function which multiplies 12 by 10. *(1 mark)* [3]

2. (a) The map function takes a list and applies a function to each element, returning a new list.

(b) map (\*3) [1,2,3,4,5]

3. isEven n = n `mod` 2 == 1

filter (isEven)[1,2,3,4,5,6]

Chapter 71 – Lists in functional programming

# In-text questions

**Q1**: Head is 3. Tail is the list [7,14,83,2,77]

**Q2**: tail (tail (tail (tail (tail names)))) returns [“George”]

Q3: (a) let newNumbers = [1,2,3,4]

or, newNumbers = [1,2,3,4]

or, 1:2:3:4:[]

(b) newList = numbers ++ newNumbers

Note that: newList = [Numbers] + [newNumbers] will produce a list of two lists:

[[3,7,14,83,2,77],[1,2,3,4]] hence tail newList in this instance would output: [[1,2,3,4]]

(c) filter (>10) newList

# Exercises

1. (a) [fox, deer, badger, seal, dolphin] [1]

(b) deer [2]

(c) True [2]

2. (a) [112, 156, 90, 124, 136] [1]

(b) [56,78,62,68] [1]

(c) [156, 124, 136] [2]

3. (a) [2,6,8] ++ xs

or 2:6:8:xs [1]

(b) xs ++ [12,13,14] [1]

(c) tail xs [1]

(d) 12:13: tail(tail xs))

or [12,13]++(tail(tail xs)) [3]

Chapter 72 – Big Data

# In-text questions

**Q1**: 3 - Lucy, Paul, Angie

**Q2**: Nodes are the round circles

Edges are the lines connecting nodes labelled with relationships.

Properties are the contents of nodes e.g. name = Roger, etc.

# Exercises

1. Functional programs are stateless (1), have no side effects (1). This means that calling a function will always produce the same result, given the same arguments.

Variables are immutable (1). All these features make it ideal for parallel processing (1) and make it easier to write correct code.

2. Dataset is very large (1) and is typically distributed among many servers (1). New nodes being added to the data with new connections does not alter the structure of a graph, but would require changes to the structure of a database as new relationships were added.

Data is highly interconnected (1) and it would be prohibitively time consuming (1) to analyse connections in a traditional database.

Data is not organised (1) and cannot easily be put into the rows and columns of a relational database (1)

3. In business, analysing the connections between customers’ purchases enables stores to make recommendations to other customers (e.g. Amazon), stock appropriate goods, position goods for maximum sales in stores.

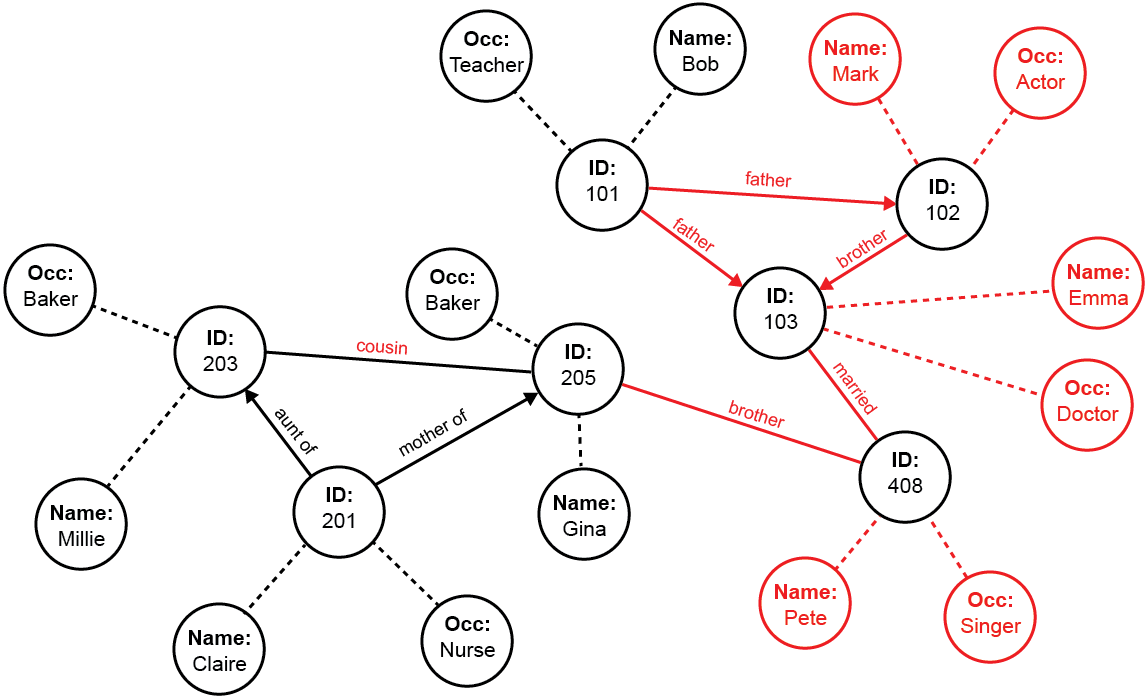
In medical applications, making connections between outbreaks of disease (e.g. flu) and searches made on Google allow health authorities to predict where an outbreak is occurring more rapidly and take appropriate action (e.g. vaccination)

In sat nav systems, weather data combined with traffic data enables the system to recommend the best route to avoid hold-ups.

In oil exploration, connections between core samples and previous oil discoveries allow predictions of the best places to drill.

(or any other suitable examples)

4.



Appendix A

Normalisation of floating point numbers, errors, range and precision

# In-text questions

**Q1**: (a) 10.5

(b) 11.625

(c) 7.375

(d) 10.875

**Q2:** (a) 00101100

(b) 10101110

(c) 01111001

(d) 00110111

**Q3:** Largest is 11111111 which is 4191.9375 Smallest is 0

**Q4:** (a) 0.1101010 x 24 = 01101.010 = 13.25

(b) 0.1001100 x 23 = 100.1100 = 4. 75

**Q5:** Mantissa 1110 = -2. (Flip all the bits and add one)

0.1100000 x 2-2 = .0011 = .125 + .0625 = .0.1875

**Q6:** (a) 0.1000000 x 2-2  = .001 = 0.125

(b) - 0.1101000 x 24 = - 1101.000 = -13.0

(c) 1.0011000 x 2-1

Take 2s complement of mantissa: -0.1101000

Move binary point one place left: -0.01101000 (since exponent = -1

= -( 0.25 + .0.125 + .03125) = -0.40625

**Q7:** (a)0.0000110 0001 = 0.0000110 0001 x 21

= 0.1100000 x 21-4

Exponent = -3.

3 = 0011. Take 2s complement, or work it out as -8 for the sign bit, -3 = 1101

Normalised binary number is 0.1100000 1101

(b) 1.1110011 0011 = 1.1110011 x 23

= 1.00110000 x 23-3

Exponent = 0

Normalised binary number is 1.0011000 0000

**Q8:** (a) 010000.11 = 0.1000011 x 25 = 01000011 0101

(b) 4.5 = 0100.1000

-4.5 = 1011.1000 = 1.0111000 x 23 = 10111000 0011

**Q9**: (a) The nearest approximation to 0.1 is 0.0625 + 0.03125 + 0.0078125

= 0.1015625 (in 8 bits)

Absolute error = 0.1015625 - 0.1 = 0.0015625

Relative error = 0.0015625/0.1 = 0.015625 or 1.5625%

(b) The nearest approximation to 0.3333333 is

0.25 +0.0625 + 0.015625 + 0.0078125 = 0.3359375

Absolute error = 0.3359375 – 0.3333333 = 0.0026042

Relative error = 0.0026042/0.3333333 = 0.007813 or 0.7813% (approx.)

# Exercises

1. (a) (1 mark for correct mantissa, 1 mark for correct exponent)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |  | 1 | 0 | 0 | 0 |

Mantissa Exponent [2]

(b) Mantissa = -0.625 // -5/8

Exponent = 2

Answer = -2.5 //-2 ½ [2]

(c) (a) (1 mark for correct mantissa, 1 mark for correct exponent)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |  |  | 0 | 1 | 0 | 0 |

Mantissa Exponent [2]

(d) Maximises precision/accuracy for given number of bits

Unique representation of each number//simpler to test for equality of numbers [2]

(e) Reduced precision

Increased range

or: can represent larger/smaller numbers

No effect on amount of memory required to represent a number. [2]

Appendix B

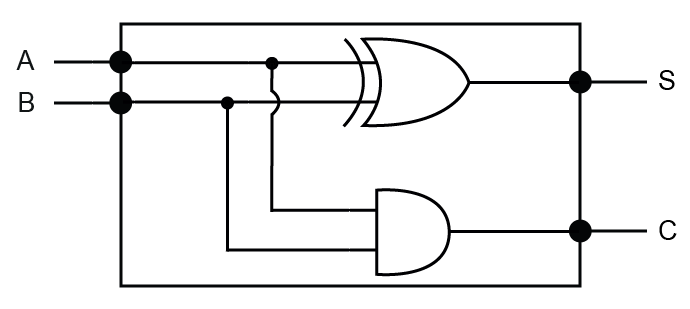
Adders and D-type flip-flops

# In-text questions

**Q1**: S4 = 1, C5 = 0.

# Exercises

1. (a)



|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **S** | **C** |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

[1]

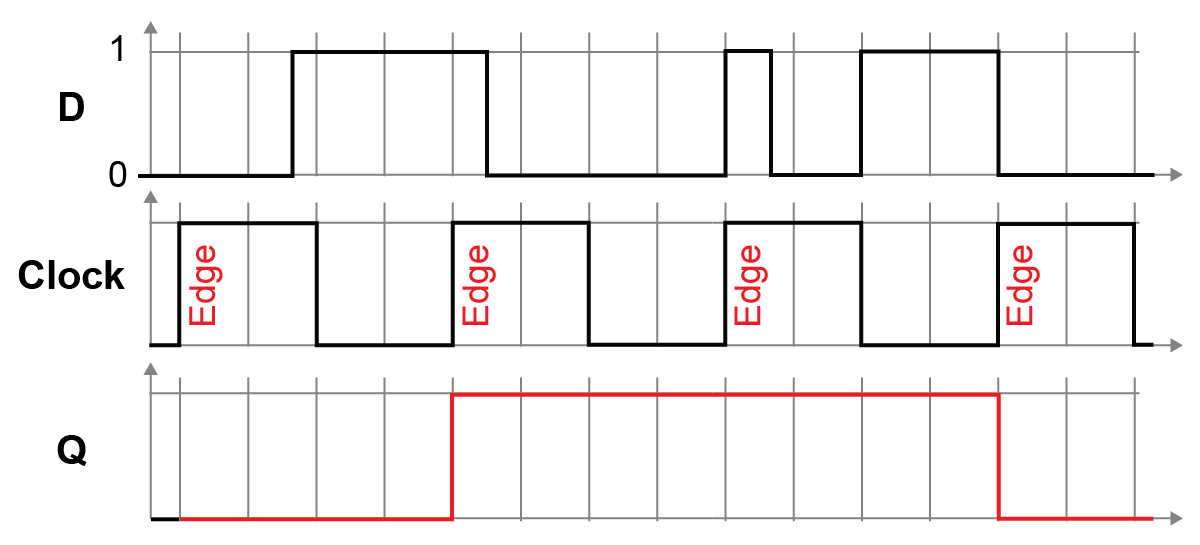
(b)

[2]

(c) A full adder has an additional input; which is the carry bit from a previous addition. [2]

1. (a) In red on diagram labelled ‘Edge’. [1]

(b) Award one mark per rising edge. [4]



Acknowledgements

The authors and publisher would like all contributors for their kind permission to reproduce their photographs or images, screenshots of their websites or other copyright material in the Answers Pack.

Answer diagrams copyright © AQA

This material contains links to relevant websites. Every effort has been made to ensure that at the time of distribution, the links remain unbroken, the material remains up-to-date and that links are not inadvertently linked to sites that could be considered offensive. PG Online cannot be held responsible for the content of any website mentioned in this material. It is sometimes possible to find relocated sites by typing the original URL into a browser. Any errors should be reported directly to support@pgonline.co.uk and changes will be made in any subsequent editions of the material.

Artwork



South Coast Sailing © 2014 Heather Duncan

Oil on canvas, 60x60cm

www.heatherduncan.com