# Final assessment test Answers

1. A hospital has asked for a program that will control the order in which the arrivals are seen in the accident and emergency department.

 (a) A priority queue has been chosen to store the arrival information.

 (i) State the property of a priority queue that makes it suitable for this application. [1]

Items are added in order of importance / items are added into different slots in the queue based on importance (priorities), because some accidents are more important than others

 (ii) The priority queue is implemented using a circular queue.

 State an advantage of a circular queue which makes it more efficient in use of memory. [1]

 1 mark = when items are removed from a circular queue, the space can be reused, thereby making more efficient use of memory space

 (b) A test version of the priority circular queue consists of 5 cells. Arrival information consists of a severity rating and a last name. The severity scale is 3 (least severe), 2 (moderately severe), and 1 (most severe). For example, Mr Jones has a severity rating of 2. His arrival record would be ‘2 Jones’.

 Complete the diagram to show the results after the following operations. [4]

 Note: The first element in the queue has an index of 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **arrivals** |  |  |  |  |
|  | **[0]** | **[1]** | **[2]** | **[3]** | **[4]** | **count** | **front** | **rear** | **returned** |
|  | 1 Black | 1 Singh | 2 Jones | 3 Reeve |  | 4 | 0 | 3 |  |
| Dequeue | 1 Black | 1 Singh | 2 Jones | 3 Reeve |  | 3 | 1 | 3 | Black |
| Enqueue ‘3 Mason’ | 1 Black | 1 Singh | 2 Jones | 3 Reeve | 3 Mason | 4 | 1 | 4 |  |
| Enqueue ‘2 Ing’ | 3 Mason | 1 Singh | 2 Jones | 2 Ing | 3 Reeve | 5 | 1 | 0 |  |
| Dequeue | 3 Mason | 1 Singh | 2 Jones | 2 Ing | 3 Reeve | 4 | 2 | 0 | Singh |

1 mark for each correct row

1. A stack is an important data structure in computer science.

 (a) Define the term stack. [2]

 A stack is a last in first out data structure where new items are added to the top of the stack (1 mark) and items are removed from the top of the stack (1 mark).

 (b) A call stack is used when subroutines are called.

 List 3 items that each call stack frame could hold. [3]

Parameters

Return address

Local variables

 (c) Explain how a recursive algorithm could cause a system to crash. [2]

A recursive algorithm calls itself as a subroutine, so will use a stack frame on every call.

If the recursive algorithm never unwinds, then all of the memory allocated to the system stack will be used up. This will cause the system to crash.

(d) State another use of a stack in a computing system. [1]

Holding calculations (reverse-polish notation)

Back buttons in web browsers

Undo buttons in editor

1. In a very busy coffee shop, orders are split between different production lines. One line only makes hot drinks. Another line only makes cold drinks. When an order arrives, items are split out and assigned to different lines. Items are given to customers as soon as the order is complete. An abstract data type **list** is used to hold the order IDs.

 (a) Describe the difference between a dynamic and a static data structure. [2]

Static data structure is fixed in size

Dynamic data structure can grow and shrink

 (b) Identify a check that should be done before removing an item (popping) from a list. [1]

 Check if the list is empty with isEmpty() or check if there is an item in the list.

 (c) Identify 2 pieces of information from the scenario that are important in deciding whether to implement the order lists as dynamic or static data structures. [2]

The list will change often because customers come and go. (A dynamic data structure will never be “full” as it can increase and shrink as items are added and deleted. A static data structure has to have its size declared in advance.)

The list is ordered numerically by order ID. (The order IDs will be added to the end of the data structure as they arrive in sequence. This cannot be done with a static data structure such as an array because you cannot add or delete items, only change a specific item.)

Completed orders need to be removed, so “holes” may need to be accounted for. (In a dynamic data structure, a specific item can be removed from the data structure and the other items will automatically adjust position so that there are no empty elements in the list. The implementation of this will vary.)

1. Hashing is often used when large amounts of data need to be stored.

 (a) Explain how a hashing algorithm works. [3]

 An algorithm is applied to the primary key (1 mark) to generate an address (1 mark) where the data is stored (1 mark).

(b) A particular hashing algorithm sums all the individual digits in a number and takes the remainder after division by 11 as the address in the hash table. Apply this algorithm to 7823901. [1]

 7+8+2+3+9+0+1=30 mod 11 = **8**

(c) In the following hash table, collision resolution is done by storing the item in the next free slot.

|  |  |
| --- | --- |
| [0] | Farmer  |
| [1] |  |
| [2] |  Jones |
| [3] | Wilson |
| [4] | Smith |
| [5] | Brown |
| [6] | Redmond |
| [7] |  |
| [8] | Green  |
| [9] |   |
| [10] | Gold |

 (i) The value “Redmond” needs to be added to the table. The hash value is 6.

 State the value of the table index where “Redmond” is stored. [1]

 6

 (ii) The value “Farmer” needs to be added to the table. The hash value is 10.

 State the value of the table index where “Farmer” is stored. [1]

 0

(d) The performance of a particular large hash table has degraded as the table fills up.

 (i) Explain why this might be the case. [1]

 As more collisions occur, it will take longer to find each entry or establish that it is not in the table. (1 mark)

(ii) Explain **one** possible way of improving the performance of the hash table. [2]

 **Increase the size of the table** and rehash all the entries, so that the load factor is reduced and there are **fewer collisions**.

1. Towns and the connections between them can be represented as a graph which has a set of nodes connected by edges. This representation is an abstraction.

(a) Define the term data abstraction. [2]

A logical view of the data and operations on it.

The details of how the data are actually represented are hidden from the user.

 (b) Complete the adjacency list for this graph. [2]

|  |  |  |
| --- | --- | --- |
| **A** | **🡪** | {C:7} |
| **B** | 🡪 | {A:3, D:4} |
| **C** | 🡪 | {B:9, D:1} |
| **D** | 🡪 | { } |

 

1 mark for each of 2 correct cells

 (c) Complete the adjacency matrix for this graph. [2]

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

 

One mark for each two correct rows

1. A tree is an abstract data type.

(a) Define the term **tree**. [2]

A connected, undirected graph (1) with no cycles (1)

(b) Here is a binary tree.



1. Complete the array implementation of this tree. [2]

|  |  |  |  |
| --- | --- | --- | --- |
|  | left | data | right |
| A[0] | 3 | H | 1 |
| A[1] | 2 | T | 4 |
| A[2] | 5 | R | -1 |
| A[3] | -1 | F | -1 |
| A[4] | -1 | X | -1 |
| A[5] | -1 | M | -1 |

(ii) List the nodes in the order in which they are visited for each traversal. [3]

 Pre-order: H, F, T, R, M, X

 In-order: F, H, M, R, T, X

 Post-order: F, M, R, X, T, H

(c) Add the following items to a binary tree so that it may be quickly searched to find a particular item:

 Liam, Mary, Zoe, Carla, David, Lucy, Adam [2]



1. Scalars have only magnitude. Vectors have both magnitude and direction. Mathematical operations can be used to combine vectors and scalars.

(a) Here are two vectors:

 a = (6, -1) b = (-3, -3)

1. Calculate the result of a + b. Show your working. [1]

 a + b = ((6 - 3), (-1 -3)) = (3, -4)

 (ii) Calculate the magnitude of vector b. Show your working. [1]

 Sqrt (x2 + y2) = Sqrt (9 + 9) = Sqrt (18) = 4.2

 (Give the mark for answer $\sqrt{18}$ )

(b) Vector u is defined as (4, 2).

 (i) Calculate 4u. Show your working. [1]

 4u = (4\*4, 4\*2) = (16, 8)

1. What is the effect of multiplying a vector by a scalar? [1]

 The vector is scaled in size, becoming either larger or smaller, but does not change direction.

(c) *p* = [4, 3, 2, 1] represents the count of people completing the number of laps around a field, expressed in *f* = [10, 20, 30, 40].

 Determine the total laps by calculating *p* • *f*. Show your working. [1]

 *p • f = (4\*10) + (3\*20) + (2\*30) + (1\*40) = 40 + 60 + 60 + 40 = 200*

(d) Dot product can be used to calculate even parity.

 *u* = [1, 1, 1, 1] and *v* = [1, 0, 0, 1].

 Determine the parity bit by calculating *u* • *v*. [2]

 (1\*1)+(1\*0)+(1\*0)+(1\*1) = 2
Parity bit is 2 mod 2 = 0

 One mark for dot product

 One mark for correct parity bit

Total 50 marks