# Homework 3 Programming Iteration Answers

1. Write a pseudocode algorithm using a FOR loop to read five lowercase letters and output the largest and smallest. (a is less than b). [6]

maxletter = “a”

minletter = “z”

FOR index 🡨 1 to 5

OUTPUT “Enter a letter”

letter 🡨 USERINPUT

IF letter > maxLetter THEN

maxLetter 🡨 letter

END IF

IF letter < minLetter THEN

minLetter 🡨 letter

END IF

END FOR

OUTPUT maxLetter

OUTPUT minLetter

2. Write a pseudocode algorithm that asks a user for a password. They are allowed three attempts to type the correct password, which is “Tues1212”.

If they type the correct password, output “Password accepted”, otherwise output “Password rejected”. [6]

OUTPUT (“Please enter password: ”)

password = USERINPUT

attempts = 1

WHILE password <> "Tues1212" and attempts < 3

OUTPUT("Password incorrect - please re-enter: ")

password = USERINPUT

attempts = attempts + 1

ENDWHILE

IF password == "Tues1212" THEN

OUTPUT("password accepted")

ELSE

OUTPUT("password rejected")

END IF

# 3. (a) Complete the trace table below with the values supplied.

sunshine 🡨 0

maxHours 🡨 0

minHours 🡨 100

totalSunshine 🡨 0

REPEAT

sunshine 🡨 INPUT

IF sunshine > maxHours Then

maxHours 🡨 sunshine

ENDIF

IF sunshine < minHours THEN

minHours 🡨 sunshine

ENDIF

totalSunshine 🡨 totalSunshine + sunshine

UNTIL sunshine = -1

OUTPUT “Max sunshine hours:”, maxHours

OUTPUT “Min sunshine hours:”, minHours

OUTPUT “Total sunshine hours”, totalSunshine

Test Data 2 7 3 8 -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sunshine** | **maxHours** | **minHours** | **totalSunshine** | **Output** |
| 0 | 0 | 100 | 0 |  |
| 2 | 2 | 2 | 2 |  |
| 7 | 7 |  | 9 |  |
| 3 |  |  | 12 |  |
| 8 | 8 |  | 20 |  |
| -1 |  | -1 | 19 | 8 -1 19 |

[4]

# (b) What is the problem with the algorithm above?

# The algorithm uses -1 to terminate but because it checks the termination after calculating the max, min and total sunshine, the values printed are wrong. This can be corrected by putting the first input statement before the loop is entered, and another input statement just before the end of the loop as below. [2]

# 

# (c) This time the algorithm uses an entry condition WHILE loop. Complete the trace table to see the difference between the two.

sunshine 🡨 0

maxHours 🡨 0

minHours 🡨 100

totalSunshine 🡨 0

sunshine 🡨 INPUT

WHILE sunshine <> -1

IF sunshine > maxHours THEN

maxHours 🡨sunshine

END IF

IF sunshine < minHours THEN

minHours 🡨 sunshine

ENDIF

totalSunshine 🡨sunshine + totalSunshine

sunshine 🡨 INPUT

END WHILE

OUTPUT “Max sunshine hours:”, maxHours

OUTPUT “Min sunshine hours:”, minHours

OUTPUT “Total sunshine hours”, totalSunshine

Input data: 2 7 3 8 -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sunshine** | **maxHours** | **minHours** | **TotalSunshine** | **Output** |
| 0 | 0 | 100 | 0 |  |
| 2 | 2 | 2 | 2 |  |
| 7 | 7 |  | 9 |  |
| 3 |  |  | 12 |  |
| 8 | 8 |  | 20 |  |
| -1 |  |  |  | 8 2 20 |
|  |  |  |  |  |

[2]

[Total 20 Marks]