**HL Unit 5** **– Abstract Data Structures**  
Quiz 1 – 2D Arrays

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 1** | | | |
| Objectives: | 5.1.4, 5.1.5 | Exam Reference: | Nov-17 15.a.b.c |

1. The collection WEATHER contains the temperatures that have been measured for one city

over the course of **one week**, starting on Monday and ending on Sunday. Each day,

24 readings were taken, one each hour, the first being at 00:00, the second at 01:00 and so

on. The data is stored in chronological order with the data for Monday stored in the collection

first, followed by Tuesday and so on.

1. State the total number of readings that were taken during this week. [1]

168; **[1]**

***Note:*** *Award* ***[1]*** *for “24 x 7” seen.*

1. Construct the algorithm to read this data into a 2D array, A, that would allow the

temperature on a specific day at a specific time to be accessed directly. [4]

*Award up to* ***[4 max]*** *as follows:*

*Example answer 1:*

*Award* ***[1]*** *for any nested loop.*

*Award* ***[1]*** *for the correct nested loop.*

*Award* ***[1]*** *for the correct assignment to A.*

*Award* ***[1]*** *for the correct retrieval from WEATHER.*

loop for DAY from 0 to 6

loop for HOUR from 0 to 23

A[DAY][HOUR] = WEATHER.getNext()

// A[DAY][HOUR] = WEATHER.getData()

end loop

end loop

*Example answer 2:*

*Award* ***[1]*** *for initialization of POS and correct increment within the loop*

*Award* ***[1]*** *for the correct loop.*

*Award* ***[1]*** *for the correct calculation of DAY*

*Award* ***[1]*** *for the correct calculation of HOUR*

*Award* ***[1]*** *for the correct assignment to A( correct retrieval from WEATHER )*

POS=0

loop while WEATHER.hasNext() //*accept not WEATHER.isEMPTY()*

DAY=POS div 24

HOUR=POS mod 24

POS=POS+1

A[DAY][HOUR]=WEATHER.getData() // *WEATHER.getNext()*

end loop

1. Construct the algorithm that will output the day, as a word (for example Tuesday), on

which the highest temperature was recorded. [6]

*Award up to* ***[6 max]*** *as follows:*

*Example answer 1:*

*Award* ***[1]*** *for initialization of HIGHEST*

*Award* ***[1]*** *for initialization of NAMES*

*Award* ***[1]*** *for the correct outer loop*

*Award* ***[1]*** *for the correct inner loop*

*Award* ***[1]*** *for the correct comparison and the assignment of HIGHEST*

*Award* ***[1]*** *for the assignment of MAX\_DAY within if statement*

*Award* ***[1]*** *for the correct output statement*

HIGHEST = A[0][0] //accept for example HIGHEST = -100

NAMES=[“Mon”, “Tue”, “Wed”, “Thu”, “Fri”, ”Sat”, “Sun”]

// array NAMES may contain other names, like “Monday”, “Tuesday”, etc.

loop for DAY from 0 to 6

loop for HOUR from 0 to 23

if A[DAY][HOUR] > HIGHEST

HIGHEST = A[DAY][HOUR]

MAX\_DAY = DAY

end if

end loop

end loop

output(NAMES[MAX\_DAY])

*Example answer 2:*

*Award* ***[1]*** *for initialization of HIGHEST.*

*Award* ***[1]*** *for the correct outer loop*

*Award* ***[1]*** *for the correct inner loop*

*Award* ***[1]*** *for the correct comparison and assignment of HIGHEST*

*Award* ***[1]*** *for the assignment MAX\_DAY within if statement*

*Award* ***[1]*** *for the if statement after the outer loop (accept switch statement)*

*Award* ***[1]*** *for the correct output statement (may be written within if statement)*

HIGHEST = A[0][0] //accept for example HIGHEST=-1000

loop for DAY from 0 to 6

loop for HOUR from 0 to 23

if A[DAY][HOUR] > HIGHEST

HIGHEST = A[DAY][HOUR]

MAX\_DAY = DAY

end if

end loop

end loop

if MAX\_DAY == 0 then

D = “Monday”

else if MAX\_DAY == 1 then

etc….

end if

output D