Rewarding Learning

General Certificate of Secondary Education
2022

Digital Technology

Unit 4

Digital Development Concepts

[GDG41]

MONDAY 6 JUNE, AFTERNOON

MARK SCHEME

General Marking Instructions

Introduction

Mark schemes are published to assist teachers and students in their preparation for examinations. Through the mark schemes teachers and students will be able to see what examiners are looking for in response to questions and exactly where the marks have been awarded. The publishing of the mark schemes may help to show that examiners are not concerned about finding out what a student does not know but rather with rewarding students for what they do know.

The Purpose of Mark Schemes

Examination papers are set and revised by teams of examiners and revisers appointed by the Council. The teams of examiners and revisers include experienced teachers who are familiar with the level and standards expected of students in schools and colleges.

The job of the examiners is to set the questions and the mark schemes; and the job of the revisers is to review the questions and mark schemes commenting on a large range of issues about which they must be satisfied before the question papers and mark schemes are finalised.

The questions and the mark schemes are developed in association with each other so that the issues of differentiation and positive achievement can be addressed right from the start. Mark schemes, therefore, are regarded as part of an integral process which begins with the setting of questions and ends with the marking of the examination.

The main purpose of the mark scheme is to provide a uniform basis for the marking process so that all the markers are following exactly the same instructions and making the same judgements in so far as this is possible. Before marking begins a standardising meeting is held where all the markers are briefed using the mark scheme and samples of the students' work in the form of scripts. Consideration is also given at this stage to any comments on the operational papers received from teachers and their organisations. During this meeting, and up to and including the end of the marking, there is provision for amendments to be made to the mark scheme. What is published represents this final form of the mark scheme.

It is important to recognise that in some cases there may well be other correct responses which are equally acceptable to those published: the mark scheme can only cover those responses which emerged in the examination. There may also be instances where certain judgements may have to be left to the experience of the examiner, for example, where there is no absolute correct response – all teachers will be familiar with making such judgements.

COVID-19 Context

Given the unprecedented circumstances presented by the COVID-19 public health crisis, senior examiners, under the instruction of CCEA awarding organisation, are required to train assistant examiners to apply the mark scheme in case of disrupted learning and lost teaching time. The interpretation and intended application of the mark scheme for this examination series will be communicated through the standardising meeting by the Chief or Principal Examiner and will be monitored through the supervision period. This paragraph will apply to examination series in 2021–2022 only.

1 **(a)** D

P	Q	P AND Q	NOT(P AND Q)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

[1]

(b) B

P	Q	P AND Q	R	(P OR Q) AND R
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

[1]

(c)

For x = 0 To 10 Output x * 2	A 0 2 4 6 8 10 12 14 16 18 20 [1]

X = 7, Y=12, Z=20
If X < Y Then
   Output X
ELSE
  If Y < Z and Y < X Then
     Output Y
EndIf
If X < Z Then
   Output Z
	B 7 20
	[1]

[2]

(d)

Definition	Tick(✓)
The correction of errors in a program whilst it is running.	
The filtering out of important information related to the problem to be solved	
The filtering out of information that will not be needed to solve the problem	✓
The use of error handling techniques to ensure data is correct	

[1]

(e)

Definition	Tick(✓)
Using functions within a program	
Breaking problems down into smaller problems when creating a solution	✓
Creating a program using only relevant information	
Creating a large problem from several small problems	

[1]

AVAILABLE MARKS

(f) C The binary search is more efficient than the linear search because it cuts the number of items it searches in half after every search [1]

(g) A ASCII stands for American Standard Code for Information Interchange and there are two forms of it, 7 bit and 8 bit [1]

8

2 **(a)** ONE from:
real/double/float [1]

(b) Sample answer

1. Output “Enter race time 3” [1]
2. Input racetime3 [1]
3. 1 for each correction condition Max 2 [2]
4. 1 for correct use of final Else/Else if contdition [1]
5. Any correct output statement related to the condition/Else [1]

Note that alternative solutions using one correct condition and nested if statements are acceptable if correct output is achieved. [6]

(c) **(i)**

	How can these test data sets help test the program?
Test Data Set 1 racetime1 31.0 racetime2 32.0 racetime3 36.6	1. racetime1 will be output as smallest 2. Provides a set of valid data/data within the correct range/ boundaries [1] [2]
Test Data Set 2 racetime1 37.9 racetime2 30.0 racetime3 33.9	1. Extreme case 30.0/lower boundary/reference to 30.0 2. racetime2 will be output as smallest 3. Provides a set of valid data/data with the correct range/ boundaries [1] [2]
Test Data Set 3 racetime1 41.0 racetime2 40.0 racetime3 31.9	1. Invalid data 41.0/out of range 2. Extreme value 40.0 3. racetime 3 is output as the smallest. 4. Provides a set of valid data/data with the correct range/ boundaries [1] [2]

[6]

(ii) Use null data or enter 0/enter a negative value/value less than 30/ example of another data type value, e.g. a string or alphanumeric value. [1]

14

			AVAILABLE MARKS
3	(a)	SOURCE code must be TRANSLATED into MACHINE code so that it can be understood by a computer. This is completed by a COMPILER. [4]	
	(b)	Any **two** from: Auto completion/auto indent [1] Clipboard [1] Code Completion Tools [1] Collapsible code sections [1] Colour coded text/colour coded functions [1] Do not accept colour coding on its own. IntelliSense [1]/syntax error assistance Line Numbering [1] [2]	6

4 **(a)** 1. Correctpriceassignment0.55(after>30) [1]
2. Condition1 : >15/>=16 [1]
3. Correctprice0.65 [1]
4. Condition2 : >0/>=1/Else [1]
5. Correctprice0.75 [1]
6. Calculationoftotalcost/outputoftotalcost [1]

7. Assignment statements must be structured correctly
Remember that there should only be 1 condition in each IF statement
Reverse allowed but remember only one condition on each line

SAMPLE ANSWER

```
IF numberofcupcakes>30
    totalcost=0.55 * numberofcupcakes
ELSE IF numberofcupcakes >=16
        totalcost=0.65 * numberofcupcakes
ELSE IF numberofcupcakes >=1
        totalcost=0.75 * numberofcupcakes
OUTPUT totalcost
```

OR

```
IF numberofcupcakes>30
    totalcost=0.55 * numberofcupcakes
ELSE IF numberofcupcakes >=16
    totalcost=0.65 * numberofcupcakes
    ELSE totalcost=0.75 * numberofcupcakes
OUTPUT totalcost
```

Other acceptable solution. [6]

(b) **(i)**

```
Do
    valid = TRUE [1]
    Output "Enter number of cupcakes"
    Input numberOfCupcakes
    if  numberOfCupcakes <1 [1] OR [1]  numberOfcupcakes>50 [1]
        valid = FALSE [1]
        Output ErrorMessage
    end if
WHILE valid = false
```

[5]

(ii)

Variable	Data Type
Valid	Boolean/Bool [1]
numberOfCupcakes	Integer/int/numeric [1]

[2]

				AVAILABLE MARKS
(c)	(i)	CONCAT/CONCATENATION/CONCATENATE	[1]	
	(ii)	ErrorMessageAll =[1] String concatenation of(ErrorMessage + ErrorMessage2) [1] [1] for assignment of both strings to ErrorMessageAll [1] for use of concatentation function/syntax/algorithm Note that code or algorithm is acceptable. Accept a coherent explanation which describes how the two strings are concatenated to produce a single string ErrorMessageAll [1]	[2]	
	(iii)	Length /len/STRLEN function	[1]	17

5 **Level 0 [0]**
Answer is not worthy of credit.

Level 1 ([1]–[2])
The candidate refers to one [1] or two [2] of bubble sort and insertion sort and mentions efficiency. The candidate makes limited use of spelling, punctuation and grammar. The meaning of the text is not always clear. The candidate displays a limited form and style appropriate to the question. The organisation of the answer is limited.

Level 2 ([3]–[4])
The candidate describes one [3] or two [4] of bubble sort and insertion sort and mentions efficiency. The candidate makes satisfactory use of spelling, punctuation and grammar. The meaning of the text is usually clear. The candidate demonstrates a satisfactory form and style appropriate to the question. The organisation of the answer is satisfactory.

Level 3 ([5]–[6])
The candidate fully describes bubble sort and insertion sort and correctly refers to efficiency. The candidate uses a good standard of spelling, punctuation and grammar. The meaning of the text is always clear. The candidate demonstrates a good standard of form and style appropriate to the question. The organisation of the answer is good.

Answers may include:
Bubble Sort:
works by comparing adjacent elements
swaps elements if they are out of order
a complete pass through the array is when all elements have been compared with their adjacent element
at the end of the first pass the smallest/largest value is at the bottom of the array
uses nested loops
reference to inefficiency, e.g. for large sets of numbers
reference to reducing the number of comparisons on each pass

Insertion Sort:
The first element is placed in the sorted sub list
Works by comparing a target value to the values in the sorted sublist
If the target value is larger/smaller than an item in the sorted sublist, the item is shifted
(swapped is acceptable) until it is in the correct place in the sorted sublist
each item in the list is compared and inserted into its correct place in the sorted sublist
reference to efficiency when compared to the bubble sort.
Efficient for processing partially sorted lists. Only requires one pass. [6] 6

6 (a) (i) Any **two** from:
bit [1]
the smallest unit of computer storage [1]
represented in 1s and 0s (do not accept ‘1s and 0s’ on its own). [1]
can be either 0 or 1/has two states, e.g. on off [2]

(ii) Conversion work – accept divide by two or place value [1]
00110110 [1]/allocate [1 mark for 6 bits such as 110110 [1] or 7 bits 0110110 [1] [2]

(b) (i) a nibble is 4 bits [1] e.g. 1100 1001 is 2 nibbles [1] /half a byte [1] [2]

(ii) 1100 = 12 = C [1]
1001 = 9 = 9 [1]
1 mark for 9C
maximum of 1 mark if conversion work is not shown [2]

(c) (i) Magnitude of number exceeds the value that can be represented by the computer/exceeds 8 bits/exceeds 1 byte [1]
result will be incorrect [1] [2]

(ii)

Calculation	**Overflow – Yes or No**
11100011 + 00000001	NO [1]
11100011+ 01101110	YES [1]
11100001+11111000	YES [1]

[3]

(d) 2TB = 2 × 1024 GB = 2048 GB [1]
2048 – 1987 = 61GB [1]/1987+50=2037 [1]
Yes the software will fit. [1]

Accept answers which use 1000 GB for the calculation
2TB = 2 × 1000 = 2000 GB [1]
2000 – 1987 = 13 GB [1]/1987 + 50 = 2027 [1]
No the software will not fit [1] [3] 16

7 (a) (i) Use the array/list name [1] followed by the position/index of the item [1] Suitable Example: **“traffic[2]”** is the third element in the array and is equal to 236/traffic[2] = 236 [1] [3]

(ii) SAMPLE ANSWER

```
    total cars = 0
    FOR i = 0 TO 4
        totalCars=totalCars+traffic[i]
            IF traffic[i] > 1000
               OUTPUT “Contact Principal – more than 1000 cars on day”,
               i+1 increment counter [not essential]
END FOR
OUTPUT totalCars
averageCars=totalCars/5
OUTPUT averageCars
```

APPLY THE FOLLOWING ABOVE. Any 11 from:

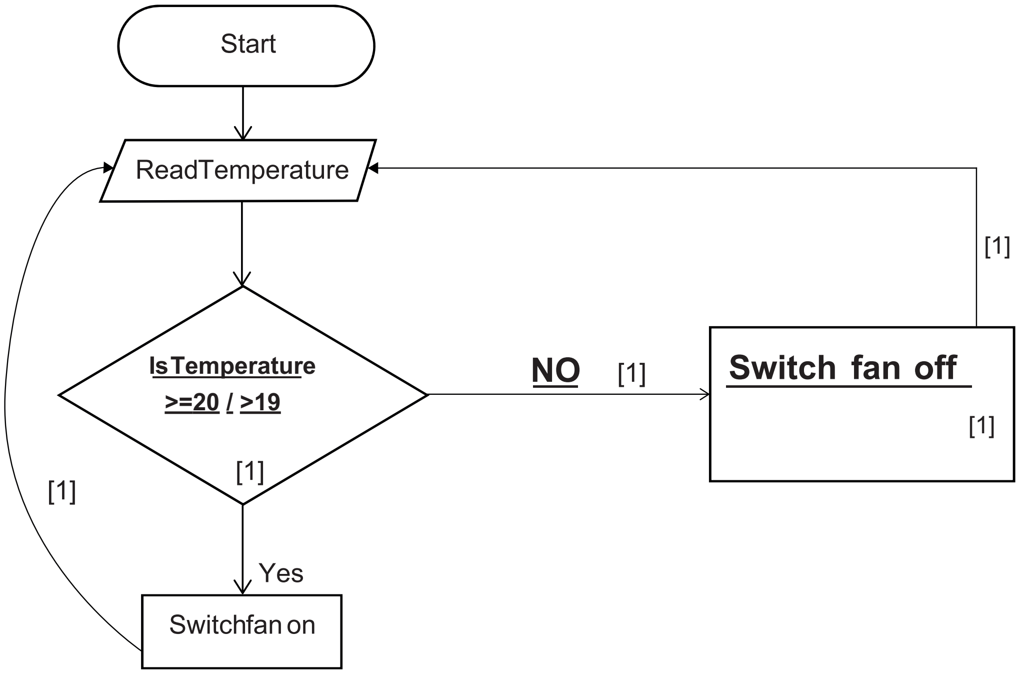
1. any initialisation/declaration [1]
2. use of loop [1]
3. loop counter to 5/len traffic/0 to 4 [1]
4. calculation of totalCars included [1]
5. correct use of traffic in calculation [as a list/array] [1]
6. correct use of i [or relevant index variable as used in the loop] as traffic[1]/index in calculation [1]
7. if statement condition (traffic[i]<1000) [1]
8. if statement inside loop [1]
9. output statement principal [1]
10. output totalCars [1]
11. averageCars calculation [1]
12. output averageCars [1] [11]

(b) (i) Any **two** from:
procedure/subroutine/block/(named) section of code/accept ‘piece of code’ for this year only [1]
written to perform a specific task [1]
can return a value/can have parameters [1]
Do not allocate marks for **benefits** as these are required for the next part of the question. [2]

(ii) Any **two** from:
helps code readability [1] do not accept ‘understand’ for this
creates re-useable code [1]/reference to enhancing code efficiency [1]
can be designed/tested individually/errors can be identified more easily [1] [2]

18

8 (a)



Do not allocate a mark where the candidate has linked to the start symbol
You can allocate 1 mark where the candidate has linked to the flow line below Start or into the ReadTemperature input symbol [5]

(b) (i) Any **two** from:
manual/paperbased run through a solution/showing results in a table [1]
which will highlight errors in logic [1]
used in whitebox testing [1] [2]

(ii) Any **two** from:
The programmer would create a (trace) table [1]
place each line of code on a separate row [1]
add conditions to columns [1]
add different values for temperature/sample values for temperature [1]
trace through each line of the code [1]
enter results into the table [1]

Diagram/Table acceptable [2]

AVAILABLE MARKS

9

9 (a) Any **two** from:
File Open[1] – used to open a file [1]
File Create [1] – used to create a new file [1]
File Append [1] used to add data to an existing file [1]
File Close [1] – Close the file [4]

(b)

Syntax error	Spelling error/suitable example [1] Undeclared variable [1] Missing symbol [1] program will not compile [1] [1]
Run-Time Error	Divide by zero [1] File does not exist [1] Array subscript out of bounds [1] Program compiles but crashes [1] [1]
Logical Error	Unexpected output from the program/ program doesn't work as planned [1] Wrong condition in loop [1] Wrong condition in IF [1] Wrong formula in calculation/suitable example, e.g. 2 + 3 = 6 [1] 'bug in code' is too vague [1]

[3] 7

10 **(a)**

Description	Test Type
Alice needs to test a function REGISTER PUPIL which records pupil details	UNIT [1]
Alice wants to check that when pupils are registered in REGISTER PUPIL they appear in the reports produced	INTEGRATION [1]
Alice wants to test the system as a whole to ensure that it meets the user requirements	SYSTEM [1]
Alice wants to test the inputs and outputs of her system to ensure they work correctly and produce the correct results	BLACK BOX [1]
Alice wants to test each line of code to ensure it works correctly	WHITE BOX [1]

[5]

(b) **(i)** Any **two** from:
(using the IDE to help) detect/correct/find/fix errors [1]
reference to stepping through the program [1]
Using exception handling [1] simple example [1]
Mention of ‘runtime’ [1]
Debugging [1]
Try Catch [1]
Stop crashing if there is an error [1] [2]

(ii) Any **two** from:
use it when developing the program [1]/use it when testing the program [1]
reference to program will run without error/will not crash [1]
reduce the amount of errors/remove errors [1] [2]

9

11 (a)

Statement	TRUE or FALSE
Evaluation of a system happens only when the system is complete.	FALSE
The end user is never involved in the evaluation process, only the programmers are involved in this process.	FALSE
To ensure that a system meets its original design criteria an evaluation must be linked to the user requirements.	TRUE
Evaluation occurs throughout the development process and is used to improve the system being developed.	TRUE

[4]

(b)

Jack wants to ensure that…	Why is this important?	Give an example from the information provided above
The system is a full and complete solution	All requirements met Meets design/plan Fully functional [1]	Register customers Make bookings Process payments List bookings List payments [1]
The system is a robust solution	The system needs be able to handle invalid [1]/high volumes of [1]/exceptional [1] data System does not crash... with qualification of answer given [1]	Any reference to managing invalid/high volume/ exceptional... examples from above [1]
The system is an efficient solution	Information is produced when required user response time acceptable (answers related to timeliness/speed – ok) reference to efficiency of code e.g. use of functions [1]	Any reference to time taken to produce reports/ register new customers/ process payments/make bookings [1]

[6]

AVAILABLE MARKS

10

Total **120**