



CANDIDATE
NAME

--

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--

0478/22

May/June 2020

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

- Answer **all** questions.
- **Do not attempt Tasks 1, 2 and 3** in the copy of the pre-release material on page 2; these are for information only.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

- The total mark for this paper is 50.
- The number of marks for each question or part question is shown in brackets [].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Blank pages are indicated.

Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A car park payment system allows customers to select the number of hours to leave their car in the car park. The customer will get a discount if they enter their frequent parking number correctly. The system calculates and displays the amount the customer must pay. The price of parking, the number of hours the customer can enter, and any discount depend upon the day of the week and the arrival time. The number of hours entered is a whole number. The price per hour is calculated using the price in force at the arrival time. No parking is allowed between Midnight and 08:00.

Day of the week	Arrival time			
	From 08:00 to 15:59		From 16:00 to Midnight	
	Max stay in hours	Price per hour	Hours	Price
Sunday	8	2.00	Up to Midnight	2.00
Monday	2	10.00	Up to Midnight	2.00
Tuesday	2	10.00	Up to Midnight	2.00
Wednesday	2	10.00	Up to Midnight	2.00
Thursday	2	10.00	Up to Midnight	2.00
Friday	2	10.00	Up to Midnight	2.00
Saturday	4	3.00	Up to Midnight	2.00

A frequent parking number can be entered for discounted parking. This number consists of 4 digits and a check digit that is calculated using a modulo 11 check digit calculation. A discount of 50% is available for arrival times from 16:00 to Midnight; the discount is 10% at all other arrival times.

Write and test a program or programs to simulate the car park payment system.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Calculating the price to park.

A customer inputs the day, the hour of arrival excluding minutes (for example 15:45 would be 15), the number of hours to leave their car, and a frequent parking number if available. If the frequent parking number has an incorrect check digit, then no discount can be applied. The price to park, based on the day, the hour of arrival, the number of hours of parking required and any discount available, is calculated and displayed.

Task 2 – Keeping a total of the payments.

Extend **Task 1** to keep a daily total of payments made for parking. The daily total is zeroed at the start of the day. For the simulation, each customer inputs the amount paid, this must be greater than or equal to the amount displayed. There is no change given so the amount input may exceed the amount displayed. Each customer payment is added to the daily total, and this total is displayed at the end of the day.

Task 3 – Making payments fairer.

Customers have complained that sometimes they are being charged too much if they arrive before 16:00 and depart after 16:00. Extend **Task 1** to calculate the price before 16:00, then add the evening charge. For example, a customer arriving at 14:45 on a Sunday and parking for five hours was previously charged 10.00 and would now be charged 6.00

1 All variables, constants and other identifiers must have meaningful names.

- (a) Identify **two** variables that you could have used for **Task 1**. Give the data type and state the use of each variable.

Variable 1

Data type

Use

.....

Variable 2

Data type

Use

.....

[4]

- (b) Data input by a customer for **Task 1** includes the day and the hour of arrival.

Identify **one** suitable validation check for each input and justify your choice. Your validation checks must be different.

Day of arrival – validation check

.....

Justification

.....

.....

.....

Hour of arrival – validation check

.....

Justification

.....

.....

.....

[4]

- (c) Explain how your program for **Task 1** ensured that the frequent parking number entered by the customer had a valid check digit.

[4]

- (d) Write an algorithm for part of **Task 2** that simulates customer payment and calculating total payments using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed.

[illegible]

- (e) Explain how your program completed **Task 3**. Assume that **Task 2** has been completed. Any programming statements used in your answer must be fully explained.

[4]

Section B

- 2** Most programming languages include basic data types. Ahmad is describing the basic data types he has used.

State the data type that Ahmad is describing in each sentence.

Choose the data type from this list of programming terms.

Array	Boolean	Char	Constant	Function	Integer
Iteration	Procedure	Real	String	Variable	

A number with a fractional part that can be positive or negative and used in calculations

Data type

A whole number that can be positive, negative or zero and used in calculations

Data type

A single number, symbol or letter

Data type

A sequence of characters

Data type

A data type with two values, True or False

Data type

[5]

- 3 (a) An algorithm has been written in pseudocode to input the names and marks of 35 students. The algorithm stores the names and marks in two arrays `Name[]` and `Mark[]`. The highest mark awarded is found and the number of students with that mark is counted. Both of these values are output.

```

01 HighestMark ← 100
02 HighestMarkStudents ← 0
03 FOR Count ← 1 TO 35
04     OUTPUT "Please enter student name"
05     INPUT Name[Count]
06     OUTPUT "Please enter student mark"
07     INPUT Mark[Counter]
08     IF Mark[Count] = HighestMark
09         THEN
10             HighestMarkStudents ← HighestMarkStudents + 1
11     ENDIF
12     IF Mark[Count] > HighestMark
13         THEN
14             Mark[Count] ← HighestMark
15             HighestMarkStudents ← 1
16     ENDIF
17 NEXT Count
18 OUTPUT "There are ", HighestMarkStudents, " with the highest mark of ",
    HighestMark

```

Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number

Correction

.....

Error 2 line number

Correction

.....

Error 3 line number

Correction

.....

Error 4 line number

Correction

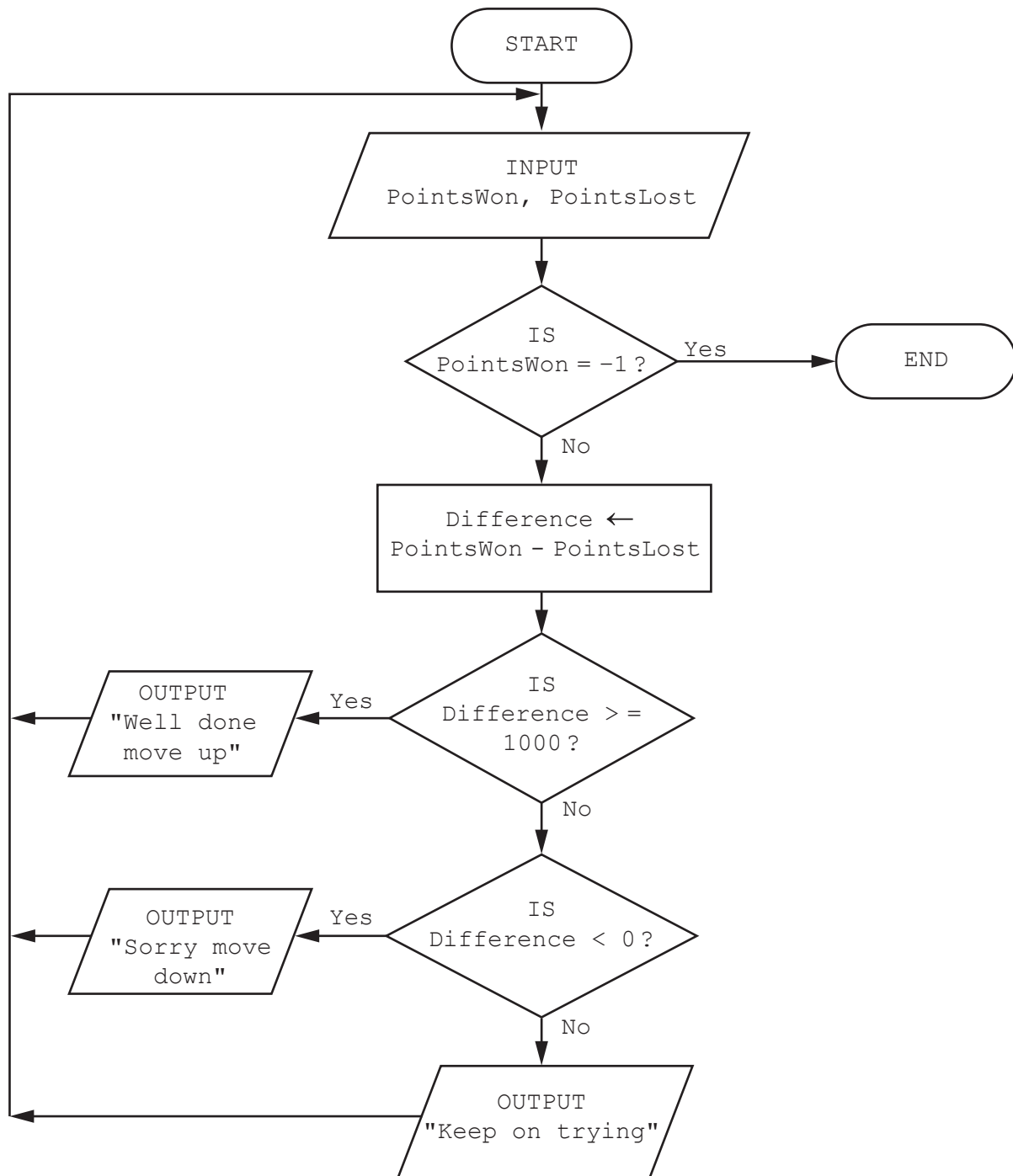
.....

[4]

- (b)** Explain how you could extend the algorithm to also find the lowest mark awarded, count the number of students with that mark, and output both these values.

[6]

- 4 This flowchart inputs the points won and the points lost when playing a game. The difference between the points won and lost is calculated and depending on the result the player can: move up to the next level, stay at the same level, or move down to the previous level. The flowchart finishes when the input for points won is -1 .



- (a) Complete a trace table for this set of input data:
5000, 4474, 6055, 2000, 7900, 9800, 3000, 2150, -1, 6700, 7615

PointsWon	PointsLost	Difference	OUTPUT

[3]

- (b) The flowchart needs to be changed. When the difference is more than 5000 the output message is 'Fantastic leap up two levels'.

Describe the changes that will need to be made to the flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

- 5 Arrays are data structures used in programming. Explain what is meant by the terms dimension and index in an array. Use examples of arrays in your explanations.

Dimension

.....

.....

.....

.....

Index

.....

.....

.....

.....

[3]

- 6 A database table, JUICE, is used to keep a record of cartons of fresh fruit juice available for sale.

Juice code	Fruit 1	Fruit 2	Size	Volume (ml)	Stock level
LMO10	Mango	Orange	Large	1000	18
MOO05	Orange	Orange	Medium	500	8
SAM02	Apple	Mango	Small	200	25
SAA02	Apple	Apple	Small	200	50
SPP02	Pineapple	Pineapple	Small	200	10
MMM05	Mango	Mango	Medium	500	12
LMM10	Mango	Mango	Large	1000	5
MGG05	Guava	Guava	Medium	500	5
SMO02	Mango	Orange	Small	200	7
MOP05	Orange	Pineapple	Medium	500	12
LAA10	Apple	Apple	Large	1000	32
SGO02	Guava	Orange	Small	200	10
LPP10	Pineapple	Pineapple	Large	1000	3
LOO10	Orange	Orange	Large	1000	25
SOO02	Orange	Orange	Small	200	40

- (a) Identify a suitable field to use as the primary key. State a reason for your choice.

Field

Reason

..... [2]

- (b) Complete the query-by-example grid to display only the stock level and size of all cartons containing only apple juice.

Field:				
Table:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

[4]

BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which itself is a department of the University of Cambridge.



Cambridge IGCSE™

COMPUTER SCIENCE

0478/22

Paper 2

May/June 2020

MARK SCHEME

Maximum Mark: 50

<p>Published</p>

Students did not sit exam papers in the June 2020 series due to the Covid-19 global pandemic.

This mark scheme is published to support teachers and students and should be read together with the question paper. It shows the requirements of the exam. The answer column of the mark scheme shows the proposed basis on which Examiners would award marks for this exam. Where appropriate, this column also provides the most likely acceptable alternative responses expected from students. Examiners usually review the mark scheme after they have seen student responses and update the mark scheme if appropriate. In the June series, Examiners were unable to consider the acceptability of alternative responses, as there were no student responses to consider.

Mark schemes should usually be read together with the Principal Examiner Report for Teachers. However, because students did not sit exam papers, there is no Principal Examiner Report for Teachers for the June 2020 series.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the June 2020 series for most Cambridge IGCSE™ and Cambridge International A & AS Level components, and some Cambridge O Level components.

This document consists of **8** printed pages.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks
1(a)	<p>For two variables One mark for identification of a variable with a suitable data type and one mark for its use</p> <p>Variable hourArrived Data type integer Use to calculate the price based on the number of hours parking</p> <p>Variable priceToPay Data type real Use store result of price calculation</p> <p>Many correct answers, these are examples only.</p>	4
1(b)	<p>For each validation check One mark for identification of a check and one mark for the justification</p> <p>Day Check presence check Justification need to enter day to be able to calculate price</p> <p>Hour Check range check Justification need to check hour is between 8 and 23</p> <p>Many correct answers, these are examples only.</p>	4

Question	Answer	Marks
1(c)	<p>Any four from:</p> <ul style="list-style-type: none"> • All 5 digits input • Each of the first 4 digits is multiplied by a weight based on its position • Specific method of individualising the digits described • Added together divided by 11 • Remainder used to calculate correct 5th digit • Checked against 5th digit input • If the same, then valid 	4
1(d)	<p>Any four from:</p> <ul style="list-style-type: none"> • Input amount • Check amount greater than or equal to the price to pay ... • ... True – add amount to daily total • ... False – ask for payment to be re-entered • ... Until valid amount input <p>Sample answer</p> <pre>PRINT "Please Enter Payment" INPUT amount WHILE amount < priceToPay DO PRINT "Insufficient payment, please enter new Payment" INPUT amount ENDWHILE totalPayment ← totalPayment + amount</pre>	4
1(e)	<p>Explanation</p> <p>Any four from:</p> <ul style="list-style-type: none"> • Add number of hours selected to arrival hour • Check if this is greater than 16 • True – calculate price up to 16 ... • ... using current price in force • ... and add in evening charge • False – calculate price as usual <p>Programming statements can be used but must be explained.</p>	4

Question	Answer	Marks
2	Real Integer Char/String String Boolean	5

Question	Answer	Marks
3(a)	One mark for error and correction Line 1 <code>HighestMark ← 0</code> Line 7 <code>INPUT Mark[Count]</code> Line 10 <code>HighestMarkStudents ← HighestMarkStudents + 1</code> Line 14 <code>HighestMark ← Mark[Count]</code>	4
3(b)	Any six from: Add variable <code>LowestMark</code> Set this to a high value for example 100 Add variable <code>LowestMarkStudents</code> Set this to zero Check if <code>Mark[Count] = LowestMark</code> True – add 1 to <code>LowestMarkStudents</code> Check if <code>Mark[Count] < LowestMark</code> True – set <code>LowestMarkStudenta</code> to 1 and set <code>LowestMark</code> to <code>Mark[Count]</code> Add extra output statement	6

Question	Answer	Marks																												
4(a)	<p>One mark for correct input (PointsWon and PointsLost) One mark for correct calculations (Difference) One mark for correct output</p> <table><tr><th>PointsWon</th><th>PointsLost</th><th>Difference</th><th>OUTPUT</th></tr><tr><td>5000</td><td>4474</td><td>526</td><td>Keep on trying</td></tr><tr><td>6055</td><td>2000</td><td>4055</td><td>Well done move up</td></tr><tr><td>7900</td><td>9800</td><td>-1900</td><td>Sorry move down</td></tr><tr><td>3000</td><td>2150</td><td>850</td><td>Keep on trying</td></tr><tr><td>-1</td><td>6700</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	PointsWon	PointsLost	Difference	OUTPUT	5000	4474	526	Keep on trying	6055	2000	4055	Well done move up	7900	9800	-1900	Sorry move down	3000	2150	850	Keep on trying	-1	6700							3
PointsWon	PointsLost	Difference	OUTPUT																											
5000	4474	526	Keep on trying																											
6055	2000	4055	Well done move up																											
7900	9800	-1900	Sorry move down																											
3000	2150	850	Keep on trying																											
-1	6700																													
4(b)	<p>Any three from:</p> <ul style="list-style-type: none">• Add extra decision box ...• ... before checking for difference greater than or equal to 1000 // change Is difference >= 1000 to >= 1000 and <= 5000• Check for difference greater than 5000• Add extra Output 'Fantastic leap up two levels'...• ... before flowline returns to input	3																												

Question	Answer	Marks
5	<p>One mark for explanation of dimension One mark for explanation of index One mark for inclusion of an example</p> <p>The dimension is the number of indexes required to access an element. The index is the position of the element in an array For example A[25] is the 25th element of a one-dimensional array.</p>	3

Question	Answer				Marks																																			
6(a)	One mark for field and one mark for reason Field Juice code Reason only unique identifier				2																																			
6(b)	<table><tr><td>Field:</td><td>Fruit 1</td><td>Fruit 2</td><td>Size</td><td>Stock level</td></tr><tr><td>Table:</td><td>JUICE</td><td>JUICE</td><td>JUICE</td><td>JUICE</td></tr><tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td></tr><tr><td>Criteria:</td><td>= "Apple"</td><td>= "Apple"</td><td></td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> <p>One mark for each correct column</p>				Field:	Fruit 1	Fruit 2	Size	Stock level	Table:	JUICE	JUICE	JUICE	JUICE	Sort:					Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Criteria:	= "Apple"	= "Apple"			or:										4
Field:	Fruit 1	Fruit 2	Size	Stock level																																				
Table:	JUICE	JUICE	JUICE	JUICE																																				
Sort:																																								
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																				
Criteria:	= "Apple"	= "Apple"																																						
or:																																								