Cambridge IGCSE™(9–1)

CANDIDATE NAME	

CENTRE NUMBER		CANDIDATE NUMBER	

COMPUTER SCIENCE **0984/22**

Paper 2 Problem-solving and Programming **May/June 2022**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- 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.

INFORMATION

- 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 **12** pages.

[Turn over

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 following tasks before the examination to answer Question 1.

Pre-release material

A program is needed to allow a Wildlife Park to sell tickets. A booking consists of one or more tickets for the same day(s) and can be made up to a week in advance. A booking can be made for a visit of one day or two consecutive days. A booking can have extra attractions included. A booking will be valid for the day(s) chosen only.

Ticket type	Cost for one day	Cost for two days
one adult	$20.00	$30.00
one child (an adult may bring up to two children)	$12.00	$18.00
one senior	$16.00	$24.00
family ticket (up to two adults or seniors, and three children)	$60.00	$90.00
groups of six people or more, price per person	$15.00	$22.50

Extra attraction	Cost per person
lion feeding	$2.50
penguin feeding	$2.00
evening barbecue (two-day tickets only)	$5.00

Write and test a program or programs for the Wildlife Park:
- Your program or programs must include appropriate prompts for the entry of data. Data must be validated on entry.
- All outputs, including error messages, 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 – displaying the ticket options and the extra attractions available
Set up your program to:
- display the options, attractions and prices for one-day tickets
- display the options, attractions and prices for two-day tickets
- show the days available for booking; assume that there are tickets available for any valid day.

Task 2 – process a booking
Extend your program for **Task 1** to:
- input the tickets and extra attractions required, then calculate the total cost of the booking
- allocate a unique booking number
- display the booking details, including the total cost and the unique booking number
- repeat as required.

Task 3 – ensuring each booking is the best value
Check that the total for each booking gives the best value and offer an alternative if this is **not** the case. For example, buying two family tickets is better than a group ticket for a group of 10 that includes four adults and six children.

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

(a) (i)   Identify **one** constant that you could have used for **Task 1**.
Give the value and use of the constant.

Constant ....

Value ....

Use ....

....
[3]

(ii)   Identify **one** variable that you could have used for **Task 2**.
Give the data type and use of the variable.

Variable ....

Data type ....

Use ....

....
[3]

(b) Explain how your program showed the days available for booking in **Task 1**.

....

....

....

....

....

.... [3]

(c) Explain how your program made sure that each booking number allocated in **Task 2** was
unique.

....

....

....

.... [2]

(d) Write an algorithm for the part of **Task 2** that inputs the tickets and extra attractions required then calculates the total cost of the booking.

Assume that the booking is for a single day.

Use pseudocode, programming statements or a flowchart.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

(e) Explain how your program decides when a family ticket is better value in **Task 3**.

Any programming statements that you include in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

.... [3]

Section B starts on page 8

 [Turn over

Section B

2 An algorithm allows a user to input their password and checks that there are at least eight characters in the password. Then, the user is asked to re-input the password to check that both inputs are the same. The user is allowed three attempts at inputting a password of the correct length and a matching pair of passwords. The pre-defined function LEN(X) returns the number of characters in the string, X

```
01  Attempt ← 0
02  REPEAT
03    PassCheck ← TRUE
04    OUTPUT "Please enter your password "
05    INPUT Password
06    IF LEN(Password) < 8
07      THEN
08        PassCheck ← TRUE
09      ELSE
10        OUTPUT "Please re-enter your password "
11        INPUT Password2
12        IF Password <> Password
13          THEN
14            PassCheck ← FALSE
15        ENDIF
16    ENDIF
17    Attempt ← Attempt + 1
18  UNTIL PassCheck OR Attempt <> 3
19  IF PassCheck
20    THEN
21      OUTPUT "Password success"
22    ELSE
23      OUTPUT "Password fail"
24  ENDIF
```

(a) Identify the **three** errors in the pseudocode and suggest a correction to remove each error.

Error 1 ....

Correction ....

Error 2 ....

Correction ....

Error 3 ....

Correction ....

[3]

(b) The algorithm includes **two** types of check on the data input.
Identify and describe each type of check.

Type of check 1 ....

Description ....

....

Type of check 2 ....

Description ....

....

[4]

(c) Give **two** sets of test data for this algorithm and a reason for choosing each set.

Each set of test data and its reason must be different.

Set 1 ....

Reason ....

....

Set 2 ....

Reason ....

....

[4]

3 (a) Describe a one-dimensional array. Include an example of an array declaration.

....

....

....

....

....

.... [3]

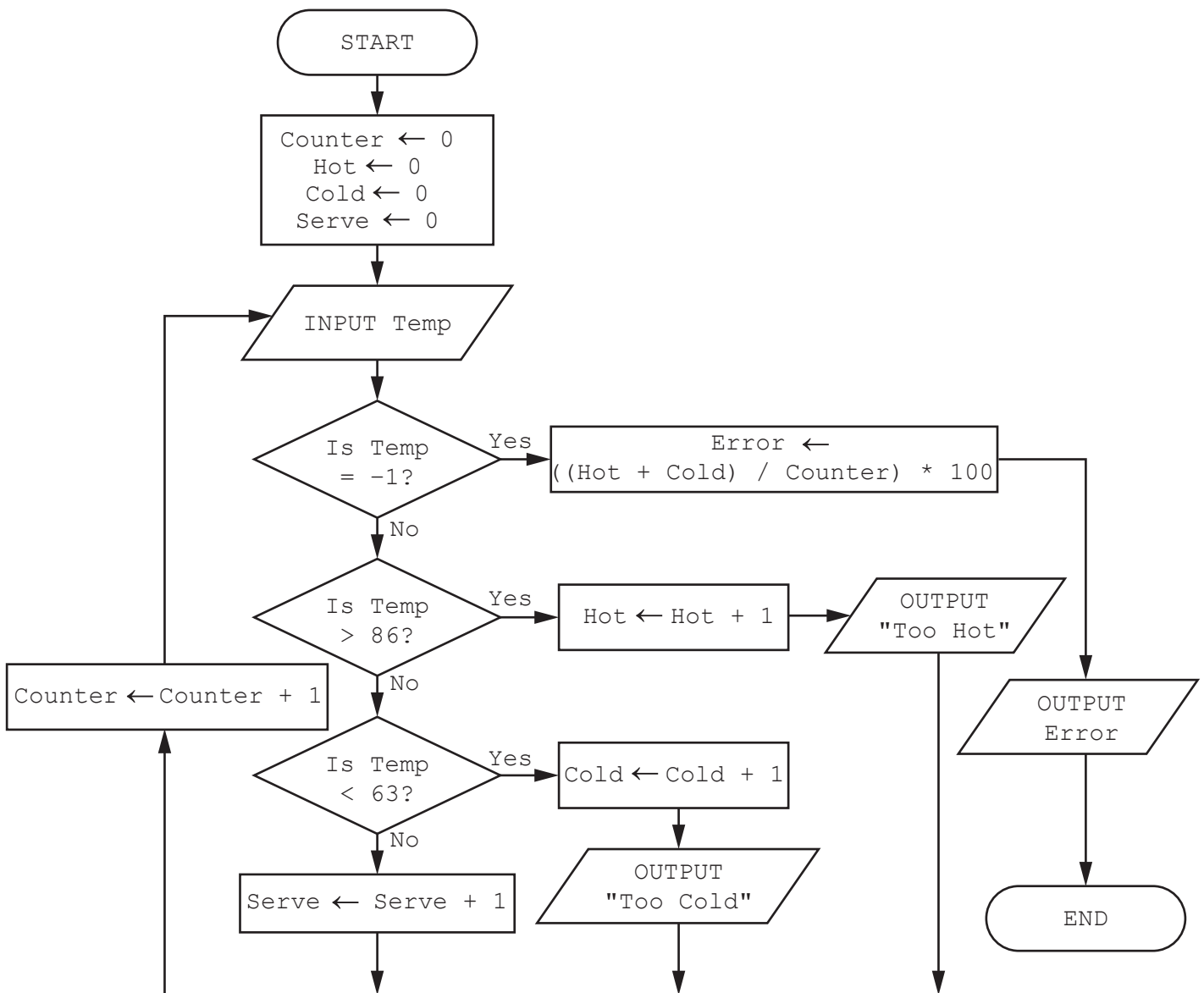**(b)** Explain how indexing could be used to search for a value stored in a one-dimensional array.

....

....

....

.... [2]

4 This algorithm checks the temperature of hot food being served to customers.

```
                          ┌──────────────┐
                          │    START     │
                          └──────┬───────┘
                                 │
                                 ▼
                    ┌──────────────────────┐
                    │   Counter ← 0        │
                    │     Hot ← 0          │
                    │    Cold ← 0          │
                    │   Serve ← 0          │
                    └──────────┬───────────┘
                               │
                               ▼
              ┌────────────────────────────┐
              │       INPUT Temp           │
              └────────────┬───────────────┘
                           │
                           ▼
                    ╱╲
                   ╱  ╲   Yes    ┌─────────────────────────────────┐
                  ╱ Is ╲─────────▶│         Error ←                 │
                  ╲Temp╱          │ ((Hot + Cold) / Counter) * 100  │
                   ╲=-1?╱         └─────────────────┬───────────────┘
                    ╲╱                              │
                    │ No                            │
                    ▼                               │
                    ╱╲                              │
                   ╱  ╲   Yes   ┌──────────────┐   ┌──────────────┐
                  ╱ Is ╲────────▶│Hot ← Hot + 1 │──▶│   OUTPUT     │
                  ╲Temp╱        └──────────────┘   │  "Too Hot"   │
      ┌───────────┐╲>86?╱                           └──────┬───────┘
      │Counter ←  │ ╲╱                                      │
      │Counter + 1│  │ No                                   │
      └───────────┘  ▼                             ┌──────────────┐
                     ╱╲                            │   OUTPUT     │
                    ╱  ╲   Yes   ┌──────────────┐  │   Error      │
                   ╱ Is ╲────────▶│Cold ← Cold + 1│ └──────┬───────┘
                   ╲Temp╱        └──────┬───────┘         │
                    ╲<63?╱              │                 │
                     ╲╱                 ▼                 ▼
                     │ No        ┌──────────────┐   ┌──────────────┐
                     ▼           │   OUTPUT     │   │    END       │
              ┌──────────────┐   │  "Too Cold"  │   └──────────────┘
              │Serve ← Serve+1│  └──────────────┘
              └──────────────┘
```

(a) Complete the trace table for the algorithm using this input data:

75, 78, 84, 87, 91, 80, 75, 70, 65, 62, –1, 20

Counter	Hot	Cold	Serve	Temp	Error	OUTPUT

[7]

(b) State how the final output from the algorithm could be improved.

....

.... [1]

(c) Identify the process in the algorithm that is **not** required.

....

.... [1]

[Turn over

5 A database table, NURSE, is used to keep a record of disposable items worn by veterinary nurses.

This is part of the table:

ItemNumber	Description	SingleUse	Uses	StockLevel	ReorderLevel
DIG1	Glove (pair)	Y	1	500	800
DIA1	Apron	Y	1	700	800
DIM5	Hair net	Y	1	650	500
DIA2	Apron	N	5	25	100
DIS4	Suit	N	3	70	50
DIV9	Shoe cover (pair)	Y	1	400	250

(a) Complete this query-by-example grid to display only the item number and the description of single use items, where the stock level is below the reorder level.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[4]

(b) Give a reason why the field SingleUse is **not** required in the table NURSE.

....

.... [1]