

Surname	Centre Number	Candidate Number
First name(s)		2



GCE AS

B500U10-1



TUESDAY, 16 MAY 2023 – AFTERNOON

COMPUTER SCIENCE – AS component 1

Fundamentals of Computer Science

2 hours

For Examiner's use only		
Question	Maximum Mark	Mark Awarded
1.	6	
2.	6	
3.	4	
4.	3	
5.	8	
6.	7	
7.	9	
8.	9	
9.	8	
10.	8	
11.	11	
12.	5	
13.	6	
14.	10	
Total	100	

ADDITIONAL MATERIALS

A calculator.

INSTRUCTIONS TO CANDIDATES

Use black ink or black ball-point pen.
Do not use gel pen or correction fluid.

Write your name, centre number and candidate number in the spaces at the top of this page.

Answer **all** questions.

Write your answers in the spaces provided in this booklet.
If you run out of space, use the additional page(s) at the back of the booklet, taking care to number the question(s) correctly.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

You are reminded of the need for good English and orderly, clear presentation in your answers.

The total number of marks available is 100.



JUN23B500U10101

Answer **all** questions.

1. Complete the following table giving **three** different data types and their storage requirements.

The first row has been completed for you.

[6]

Data Type	Storage Requirements (in bits)
Unsigned Long Integer	32



2. (a) State each step showing how a dual-core processor would process the following calculation. [4]

$$Y = (6 \times 5) + (3 \times 4) + (5 \times 9) + (2 \times 7)$$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (b) State, giving an example, why this calculation cannot be processed in fewer steps. [2]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



3. Explain data fragmentation and why the functional characteristics of a solid state drive (SSD) mean that its performance is not affected by fragmented data. [4]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



4. The following data is stored in a 4-bit register:

1	1	0	0
---	---	---	---

Demonstrate how the state of the second least significant bit can be discovered using a logical operation. [3]

.....

.....

.....

.....

.....

.....

.....

.....

.....



5. (a) Many high-level languages provide programmers with a suite of standard functions. Explain, giving an example, the possible benefits of standard functions. [4]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (b) Other than using a high-level language to write applications, a programmer might use an assembly language. Describe the advantages of using an assembly language over a high-level language. [4]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



7. In a certain computer system, real numbers are stored in normalised floating-point form using two's complementation, a 10-bit mantissa and a 4-bit exponent.

(a) Determine the closest possible denary representation of 8.6_{10} using this system. [4]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(b) Calculate the absolute and relative error that has occurred in 7(a).
State how this floating-point system could be modified to allow a more accurate representation of 8.6_{10} . [5]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



BLANK PAGE

**PLEASE DO NOT WRITE
ON THIS PAGE**



8. A sequential file is used to store a record that includes a student's ID and their total mark in a class test.

The following algorithm adds a new record to the file `gradesFile`.

```

1  Start Procedure AddRecord
2  studentID is integer
3  studentTotal is integer
4  newID is integer
5  newTotal is integer
6  inserted is Boolean
7
8  open gradesFile for input      {opens the grades file with the
                                original records}
9  open tempFile for output      {opens a temporary file to eventually contain
                                the original records and the new record}
10 output "Enter ID and studentTotal for the new Student:"
11 input newID
12 input newTotal
13 set inserted = FALSE
14
15 while (NOT EOF(gradesFile)) AND (inserted = FALSE)  {EOF – End of File}
16     read gradesFile, studentID, studentTotal
17
18     if newID < studentID then
19         write tempFile, newID, newTotal
20         set inserted = TRUE
21     end if
22
23     write tempFile, studentID, studentTotal
24
25
26 end while
27
28 if inserted = FALSE then
29     write tempFile, newID, newTotal
30 end if
31
32 while NOT EOF(gradesFile)
33     read gradesFile, studentID, studentTotal
34     write tempFile, studentID, studentTotal
35 end while
36
37 close gradesFile, tempFile
38 copy tempFile onto gradesFile {replaces the original gradesFile with the
                                updated tempFile}
39 End Procedure

```



10. Describe **two** applications that use different modes of operation.

[8]

APPLICATION 1

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

APPLICATION 2

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



BLANK PAGE

**PLEASE DO NOT WRITE
ON THIS PAGE**



11. The following algorithm is intended to sort integers stored in `myArray` into descending order.

```
1  Start Procedure Sort
2  declare myArray (0 to 6) as integer
3  n is integer
4  temp is integer
5  swapped is boolean
6
7  set n = 7
8
9  repeat
10     set swapped = FALSE
11     for i = 0 to (n - 1)
12         if myArray[i] > myArray[i + 1] then
13             temp = myArray[i + 1]
14             myArray[i + 1] = myArray[i]
15             myArray[i] = temp
16             swapped = TRUE
17         end if
18     next i
19 until (swapped = FALSE)
20
21 End Procedure
```

- (a) The algorithm does not work as intended. Identify the error and suggest a suitable change to the algorithm. [2]

.....

.....

.....

.....



(b) Identify and describe the purpose of **two** different types of repetition used in this algorithm. [4]

Repetition Type 1

Lines: **to**

.....
.....
.....

Repetition Type 2

Lines: **to**

.....
.....
.....

(c) Identify and describe the use of selection in this algorithm. [2]

Lines: **to**

.....
.....
.....
.....

(d) State an alternative type of sorting algorithm and describe the characteristics of your chosen type of sort. [3]

.....
.....
.....
.....
.....
.....
.....
.....
.....



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

END OF PAPER



BLANK PAGE

**PLEASE DO NOT WRITE
ON THIS PAGE**

