Chhetri Academy

chhetriacademy.com

GCSE COMPUTER SCIENCE

1.2

Edexcel 1CP2 | Specification point 1.2.6

How standard algorithms (bubble sort, merge sort, linear search, binary search) work

Untiered

Past paper question booklet

3	2020-2025	Untiered
question groups	available papers	tier

Before you begin

1. Show working and use units where appropriate.
2. Use extra paper if you need more space.
3. Print at actual size (100%) for measurement questions.

Name _____ **Date** _____

Whole questions are retained for context; some parts cover other points. Legacy sources are labelled.

6 Joe plays online games using the world wide web.

(a) A sequence of processes is required to open a web page.

Describe this sequence of processes.

(4)

(b) The web page uses a merge sort to display high scores.

Describe how a merge sort works.

(4)

(c) A bubble sort is carried out on this list.

5	2	4	1	9	3	7

(i) State the number of comparisons that will be made on the first pass.

(1)

(ii) State the number of swaps that will be made on the first pass.

(1)

(iii) State the number of passes that will be made.

(1)

(iv) State the condition that will cause the algorithm to end.

(1)

(d) A sorted data set contains millions of items.

State why a binary search algorithm would be preferable to a linear search algorithm for use with this data set.

(1)

(Total for Question 6 = 13 marks)

1 Computational thinking

(a) Identify the term that means breaking a problem or solution down into smaller parts.

(1)

- ☒ **A** Abstraction
- ☒ **B** Computation
- ☒ **C** Decomposition
- ☒ **D** Evaluation

(b) State **two benefits** of subprograms.

(2)

1 .....

.....

2 .....

.....

(c) Here is an algorithm that uses colours.

```
1  # --------------------------------------------------------
2  # Global variables
3  # --------------------------------------------------------
4  theColours = ["Green", "Blue", "Yellow", "Red", "Purple"]
5  colour = ""
6
7  # --------------------------------------------------------
8  # Main program
9  # --------------------------------------------------------
10
11 for item in theColours:
12     print (item)
13
14 colour = input ("Enter a colour: ")
15 while (colour != ""):
16     if (colour == "Green"):
17         print ("Green is my favourite colour")
18     else:
19         print (colour + " is a good colour")
20
21     colour = input ("Enter a colour: ")
```

(i) Give the first line number of a condition-controlled loop.

(1)

(ii) Give the first line number of iteration over every item in a data structure.

(1)

(iii) Give the line numbers of a selection.

(1)

(d) Programs can have syntax errors and runtime errors.

(i) Define the term 'syntax error'.

(1)

(ii) Runtime errors happen when a program is executing.

Explain a runtime error.

(2)

(e) Algorithms use relational and arithmetic operators.

(i) Here is a relational operator used in a conditional test.

```
count > index
```

State the **two** different results of evaluating a conditional test.

(2)

1 .....

2 .....

(ii) Identify the result of 5 // 2

(1)

- ☒ **A** 0.5
- ☒ **B** 1
- ☒ **C** 2
- ☒ **D** 2.5

(f) Programmers consider algorithm efficiency when they write code.

(i) Sorting and searching use algorithms.

Complete the table with the name of a search algorithm and a sort algorithm.

(2)

Algorithm type	Characteristic	Algorithm name
Search	Is a divide and conquer algorithm	
Sort	Is **not** a divide and conquer algorithm	

(ii) Explain **one** effect the number of comparisons has on the execution time of a sorting algorithm.

(2)

.....

.....

.....

.....

(Total for Question 1 = 16 marks)

5 Computational thinking

(a) Here is a partially completed truth table.

The logical expression for **Q** is complete.

Complete the truth table to give the:

- logical expression for **S**
- logical expression for **T**
- values for **Q**

(3)

A	B	C	S	T	Q
					NOT S AND T
0	0	0	0	0	
0	0	1	0	1	
0	1	0	0	1	
0	1	1	0	1	
1	0	0	0	0	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	1	1	

(b) Here is an algorithm to process sales data.

```
#  ------------------------------------------------
#  Global variables
#  ------------------------------------------------

salesData = [[9, 6, 32],
             [20, 12, 9],
             [9, 8, 8],
             [32, 24, 29],
             [7, 20, 12],
             [6, 9, 6]]
total = 0
currMax = 0

#  ------------------------------------------------
#  Main program
#  ------------------------------------------------
for sale in salesData:
    total = 0
    for item in sale:
        total = total + item
        if (item < currMax):
            currMax = item
    print (sale, total)
print (currMax)

```

(i) The data structure `salesData` is shown as a table of records, with fields in each record.

Complete the table with the line number for each traversal.

(2)

	Line number
Traverses the records	
Traverses the fields	

(ii) There is a logic error in the code.

Write the corrected line of code.

(1)

.....

.....

(c) Here is an algorithm.

```
1  #  ----------------------------------------------------
2  #  Constants
3  #  ----------------------------------------------------
4
5  NUM = 6
6
7  #  ----------------------------------------------------
8  #  Global variables
9  #  ----------------------------------------------------
10
11 total = 0
12
13 #  ----------------------------------------------------
14 #  Main program
15 #  ----------------------------------------------------
16
17 for number in range (1, NUM + 1):
18     remainder = NUM % number
19     if (remainder == 0):
20         total = total + number
21 print (total)
22
```

Complete the trace table.

You may not need to fill in all the rows in the table.

(4)

number	remainder	total	Output
		0	

(d) The efficiency of algorithms can be analysed.

The worst case is when the most comparisons are required.

The best case is when the fewest comparisons are required.

(i) Explain the worst case for a linear search on an array.

(2)

(ii) Explain the best case for a bubble sort on an array.

(2)

(e) A binary search algorithm looks for a target in a sorted array.

Here is an array of numbers.

Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
Value	11	22	33	44	55	66	77	88	99	100	101

Complete the table to show the steps of a binary search algorithm to look for the number **78**, which is not in the array.

You must show the calculation of the mid-point.

You may not need to fill in all the rows in the table.

(6)

Start index	End index	Calculation of mid-point	Found Y or N	Discard lower, higher or none

(Total for Question 5 = 20 marks)