0 1 **Table 1** lists some terms that are important in the theory of computation. These terms are described in **Table 2**.

Table 1

Label	Term
A	Information hiding
B	Procedural abstraction
C	Functional abstraction
D	Data abstraction
E	Problem abstraction
F	Decomposition
G	Composition
H	Automation

Complete **Table 2** by filling in the unshaded cells with the appropriate labels **(A)** to **(H)** from **Table 1**. On each row write the label that most closely matches the description.

Table 2

Description of term	**Label: A – H**
Breaking a problem into a number of sub-problems	
Models are put into action to solve problems	
Combining procedures into compound procedures	
Details are removed until the problem is represented in a way that is possible to solve because the problem reduces to one that has already been solved	

Copy the contents of all the unshaded cells in **Table 2** into your Electronic Answer Document. **[4 marks]**

0 2 The algorithm, represented using pseudo-code in **Figure 1**, describes a method to test whether an integer greater than 2 is prime or not.

Figure 1

```
INPUT Number
Root ← 1
WHILE (Root * Root) < Number
  Root ← Root + 1
ENDWHILE
d ← 2
FactorFound ← FALSE
WHILE (FactorFound = FALSE) AND (d <= Root)
  r ← Number MOD d
  IF r = 0 THEN
    FactorFound ← TRUE
  ENDIF
  d ← d + 1
ENDWHILE
IF FactorFound = FALSE THEN
  OUTPUT "Prime"
ELSE
  OUTPUT "Not prime"
ENDIF
```

The `MOD` operator calculates the remainder resulting from an integer division, for example, `10 MOD 3 = 1`.

0 2 . 1 Complete **Table 3** by hand-tracing the algorithm in **Figure 1**. Use `5` as the input value. You may not need to use all the rows in **Table 3**.

Table 3

Number	Root	d	FactorFound	r	Output

Copy the contents of all the unshaded cells in **Table 3** into your Electronic Answer Document.

[3 marks]

0 2 . 2 Complete **Table 4** by hand-tracing the algorithm in **Figure 1**. Use `25` as the input value. You may not need to use all the rows in **Table 4**.

Table 4

Number	Root	d	FactorFound	r	Output

Copy the contents of all the unshaded cells in **Table 4** into your Electronic Answer Document.

[3 marks]

0 3 The algorithm, represented using pseudo-code in **Figure 1**, describes a method to rearrange three numbers in a data structure.

Figure 1

```
Numbers[0] ← 43
Numbers[1] ← 17
Numbers[2] ← 85
FOR x ← 1 TO 2
  MyValue ← Numbers[x]
  y ← x - 1
  WHILE (y > -1) AND (Numbers[y] < MyValue)
    Numbers[y + 1] ← Numbers[y]
    y ← y - 1
  ENDWHILE
  Numbers[y + 1] ← MyValue
ENDFOR
```

0 3 . 1 Complete **Table 1** by hand-tracing the algorithm in **Figure 1**. You may not need to use all the rows in **Table 1**.

Table 1

x	MyValue	y	y > -1 ? (True/ False)	Numbers[y]	Numbers[y] < MyValue ? (True/ False)	Numbers		
						[0]	[1]	[2]
						43	17	85

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

[4 marks]

0 3 . 2 What type of rearrangement does this algorithm perform?

[1 mark]

0 4 A sentinel value is a special value **after** the end of a series of data values. It is a terminator for the series of data values but is not treated as part of the series. A sentinel value is used when you do not know how many data values are in the series.

The algorithm, represented using pseudo-code in **Figure 1**, is an attempt at a method to add numbers that are input as a series terminated by the sentinel value -1

Figure 1

```
X ← 0
Result ← 0
WHILE X ≠ -1
  INPUT X
  Result ← Result + X
ENDWHILE
OUTPUT Result
```

0 4 . 1 Complete **Table 1** by hand-tracing the algorithm in **Figure 1**. You may not need to use all the rows in **Table 1**.

The first row of **Table 1** has already been completed for you.

The sequence of numbers for input is: `4, 6, 3, 2, -1`

Table 1

X	Result	Output
0	0	–

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

[3 marks]

0 4 . 2 Comment on the result of the trace and describe how the algorithm should be modified.

[2 marks]

0 5 The algorithm, represented using pseudo-code in **Figure 1**, describes a method to access numbers in the data structure, `List`, shown in **Table 1**.

Figure 1

```
SUBROUTINE A(S, X, Y)
  P ← -1
  WHILE P = -1 AND X <= Y
    Z ← (X + Y) DIV 2
    IF List[Z] = S THEN
      P ← Z
    ELSE
      IF List[Z] < S THEN
        X ← Z + 1
      ELSE
        Y ← Z – 1
      ENDIF
    ENDIF
  ENDWHILE
  RETURN P
ENDSUBROUTINE
```

The `DIV` operator calculates the whole number part resulting from an integer division, for example, `10 DIV 3 = 3`

Complete **Table 2** by hand-tracing the algorithm in **Figure 1** when the following statement is executed.

```
Result ← A(38, 0, 18)
```

You may not need to use all the rows in **Table 2**.

The first row of **Table 2** has already been completed for you.

Table 1

List

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]
2	8	12	18	25	29	36	42	49	51	57	61	68	71	79	83	84	91	97

Table 2

S	X	Y	P	Z	List[Z]
38	0	18	-1		
Result:					

Copy the contents of all the unshaded cells in **Table 2** into your Electronic Answer Document.

[5 marks]

0 6 Explain what is meant by decomposition.

[2 marks]

0 7 The algorithm, represented using pseudo-code in **Figure 2**, describes a method to rearrange four numbers in a data structure.

Figure 2

```
Numbers[0] ← 45
Numbers[1] ← 19
Numbers[2] ← 62
Numbers[3] ← 12
FOR X ← 1 TO 3
  Y ← X - 1
  N ← Numbers[X]
  WHILE Y > -1 AND N < Numbers[Y]
    Numbers[Y + 1] ← Numbers[Y]
    Y ← Y - 1
  ENDWHILE
  Numbers[Y + 1] ← N
ENDFOR
```

Complete **Table 2** by hand-tracing the algorithm in **Figure 2**.

You may not need to use all the rows in **Table 2**.

The first row of **Table 2** has already been completed for you.

Table 2

X	Y	N	Numbers			
			[0]	[1]	[2]	[3]
			45	19	62	12

Copy the contents of all the unshaded cells in **Table 2** into your Electronic Answer Document.

[5 marks]

0 8 Define the term algorithm.

[2 marks]

0 9 The **Skeleton Program** models a very simple processor. Many details of how a real processor works were removed because they did not need to be included for the purposes of the simulation.

State the computing term that best describes the concept of removing detail that is unnecessary from a problem when developing a solution.

[1 mark]

1 0

Figure 1

```
S1 ← "011101"
S2 ← "001100"
C ← "0"
R ← ""
FOR J = 0 TO 5
  X ← 5 - J
  D1 ← S1[X]
  D2 ← S2[X]
  IF C = "0" THEN
    IF D1 = D2 THEN
      S ← "0"
      C ← D1
    ELSE
      S ← "1"
    ENDIF
  ELSE
    IF D1 = D2 THEN
      S ← "1"
      C ← D1
    ELSE
      S ← "0"
    ENDIF
  ENDIF
  R ← CONCATENATE(S, R)
ENDFOR
OUTPUT R
```

The function `CONCATENATE(X, Y)` returns the string formed by concatenating the string Y to the end of string X. For example, `CONCATENATE("cat", "dog")` returns `"catdog"`.

The strings are zero index based.

Complete **Table 1** by hand-tracing the algorithm in **Figure 1**.

You may not need to use all the rows in **Table 1**.

You do not need to indicate that `C`, `D1`, `D2` and `S` are strings.

The first row of **Table 1** has already been completed for you.

Table 1

S1	S2	C	R	J	X	D1	D2	S
"011101"	"001100"	"0"	""					
				OUTPUT:				

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

[5 marks]

1 1 . 1 Explain what is meant by composition **and** give an example where composition is used in the **Skeleton Program**.

[2 marks]

1 1 . 2 Describe **two** reasons why composition is used in the **Skeleton Program**.

[2 marks]