Qu		Marks	
01	1	**All marks for AO1 (knowledge)** |	
---	---
Breaking a problem into a number of sub-problems	F
Models are put into action to solve problems	H
Combining procedures into compound procedures	G
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	E	 **1 mark** per correct label **Note:** each label must only be used once (if given more than once, ignore all occurrences) **A**. handwritten answers **A**. lower case	4

Qu		Marks	
02	1	All marks for A02 (apply)	3

Number	Root	d	FactorFound	r	Output
5	1				
	2				
	3				
		2	FALSE	1	
		3		2	
		4			Prime

1 mark for correct columns `Root` **and** `d`
1 mark for correct column `r`
1 mark for correct columns `FactorFound` **and** `Output`

Max 2 marks if any incorrect values written in table

3

I. Annotation indicating 'no value'
I. Indication of repeating values
I. Quotes
I. Case & spelling for FactorFound & Output
A. T, F instead of True False

Qu		Marks	
02	2	All marks for A02 (apply)	3

Number	Root	d	FactorFound	r	Output
25	1				
	2				
	3				
	4				
	5	2	False	1	
		3		1	
		4		1	
		5		0	
			True		
		6			Not prime

1 mark for correct columns `Root` **and** `d`
1 mark for correct column `r`
1 mark for correct columns `FactorFound` **and** `Output`

Max 2 marks if any incorrect values written in table

<table>
<tr><th>Qu</th><th></th><th>Marks</th></tr>
<tr><td>3</td><td>1</td><td>All marks for AO2 (apply)
<table>
<tr><th rowspan="2">x</th><th rowspan="2">MyValue</th><th rowspan="2">y</th><th rowspan="2">y > -1 ? (True/False)</th><th rowspan="2">Numbers[y]</th><th rowspan="2">Numbers[y] < MyValue ? (True/False)</th><th colspan="3">Numbers</th></tr>
<tr><th>[0]</th><th>[1]</th><th>[2]</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>43</td><td>17</td><td>85</td></tr>
<tr><td>1</td><td>17</td><td>0</td><td>True</td><td>43</td><td>False</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>(17)</td><td></td></tr>
<tr><td>2</td><td>85</td><td>1</td><td>True</td><td>17</td><td>True</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>17</td></tr>
<tr><td></td><td></td><td>0</td><td>True</td><td>43</td><td>True</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>43</td><td></td></tr>
<tr><td></td><td></td><td>-1</td><td>False</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>85</td><td></td><td></td></tr>
</table>
1 mark for correct x column and MyValue column;
1 mark for correct y column (0, 1, 0, -1);
1 mark for correct Boolean values in columns 4 and 6;
A. TRUE/true, FALSE/false, Yes/No, Y/N and any other suitable indicators
1 mark for final contents of Numbers correct;</td><td>4</td></tr>
<tr><td>3</td><td>2</td><td>Mark is for AO2 (analyse)

sort from largest to smallest;
NE Sort on its own
A bubble sort;</td><td>1</td></tr>
</table>

<table>
<tr><td>4</td><td>1</td><td>3 marks for AO2 (apply)

<table>
<tr><th>X</th><th>Result</th><th>Output</th></tr>
<tr><td>0</td><td>0</td><td>–</td></tr>
<tr><td>4</td><td>4</td><td></td></tr>
<tr><td>6</td><td>10</td><td></td></tr>
<tr><td>3</td><td>13</td><td></td></tr>
<tr><td>2</td><td>15</td><td></td></tr>
<tr><td>-1</td><td>14</td><td>14</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

1 mark for correct <code>X</code> column (4, 6, 3, 2, -1);
1 mark for correct <code>Result</code> column (4, 10, 13, 15);
1 mark for final <code>Result</code> value (14) and <code>Output</code> column (14);

Max 2 if any errors</td><td>3</td></tr>
<tr><td>4</td><td>2</td><td>2 marks for AO2 (analyse)

The result is wrong // The sentinel value should not have been used in the calculation;
Subtract the last value // input first value before the <code>WHILE</code> loop and swap the instructions within the <code>WHILE</code> loop // add 1 to result after loop is finished;

A. not add the value if it is the sentinel value</td><td>2</td></tr>
</table>

Qu		Marks	
5		**5 marks for AO2 (apply)**	5

S	X	Y	P	Z	List[Z]
38	0	18	-1		
				9	51
		8			
				4	25
	5				
				6	36
	7				
				7	42
		6			
Result:					-1

1 mark for each correct set of values in the correct sequence (boxed in red);

A. entries within a boxed area sharing a row (e.g. 9 and 8 on the same row).

Max 4 if any errors

Alternative Answer Layout:
1 mark for each correct set of values in the correct sequence (4 sets boxed in red, 1 set in disjointed boxes in green and shaded light grey);

S	X	Y	P	Z	List[Z]
38	0	18	-1		
				9	51
		8		4	25
	5			6	36
	7			7	42
		6			
Result:					-1

6		**2 marks for AO1 (understand)** Breaking down a/the problem into a number of sub-problems; So that each sub-problem accomplishes an identifiable task; Each of these sub-problems might be decomposed further; **Max 2**	**2**

7		**5 marks for AO2 (apply)** **1 mark** for area A correct; **1 mark** for area B correct; **1 mark** for area C correct; **1 mark** for area D correct; **1 mark** for area E correct; Award a mark if the values in an area are correct regardless of which row they are on so long as they are in the correct overall sequence in a column. **I.** duplicated values instead of blanks **Max 4** if any errors		5

X	Y	N	Numbers			
			[0]	[1]	[2]	[3]
			45	19	62	12
1	0	19				
				45		A
	-1		19			B
2	1	62				C
3	2	12				62
	1				45	D
	0			19		
	-1		12			E

8		**2 marks for AO1 (understand)** a sequence of steps (to complete a task); **R**. set that always terminates / runs in finite time;	**2**

9		**Mark is for AO1 (knowledge)** Representational Abstraction; **A.** Abstraction	**1**

Qu		Marks
10		**5 marks for AO2 (application)**

S1	S2	C	R	J	X	D1	D2	S
"011101"	"001100"	"0"	""					
				0	5	"1"	"0"	"1"
			"1"	1	4	"0"	"0"	"0"
		"0"	"01"	2	3	"1"	"1"	"0"
		"1"	"001"	3	2	"1"	"1"	"1"
		"1"	"1001"	4	1	"1"	"0"	"0"
		"1"	"01001"	5	0	"0"	"0"	"1"
		"0"	"101001"					
				OUTPUT:		"101001"		

1 mark for each correct set of values in the correct sequence (boxed in red);

I. missing quotes
I. duplicated values in a column

If, after marking according to the boxed sections, fewer than 3 marks are awarded, **1 mark** can be awarded for each of the following, up to a maximum total of 3:

Column C completely correct;
Column R completely correct;
Columns J, X, D1, D2 and S **all** completely correct;

Max 4 if any errors

5

Qu		Marks	
11	1	**Procedural composition** **1 mark for AO1 (knowledge):** Combining subroutines to form compound subroutines // a subroutine that calls other subroutines; **A.** another subroutine **1 mark for AO2 (analyse):** Several subroutines are combined to form the compound subroutine `Serving` // `FindFreeTill`, `ServeBuyer`, `UpdateStats`, `CalculateServingTime`, `IncrementTimeWaiting`, `UpdateTills`, `OutputTillAndQueueStates` are combined into one subroutine `Serving` (**Note**: 2 or more subroutines **must** be named); Several subroutines are combined to form the compound subroutine `QueueSimulator` // `ResetDataStrucutures`, `ChangeSettings`, `ReadInSimulationData`, `OutputHeading`, `BuyerArrives`, `Serving`, `TillsBusy`, `OutputTillAndQueueStates`, `OutputStats`, `UpdateTills` are combined into one subroutine `QueueSimulator` (**Note**: 2 or more subroutines **must** be named); **Max 1** **Data composition** **1 mark for AO1 (knowledge):** Combining data objects to form compound data; **1 mark for AO2 (analyse):** Several records of type `Q_Node` are combined to form the compound data structure `BuyerQ` // (three) values/data items are combined to make a `Q_Node` // `BuyerID`, `WaitingTime` and `ItemsInBasket` are combined to make a `Q_Node` // arrays;	**2**

Qu		Marks	
11	2	**2 marks for AO2 (analyse)** **<u>Procedural composition</u>** Some groups of subroutines need to be called in two places/more than one place // the group of subroutines in `Serving` need to be called during the main simulation time and also after buyers stop arriving; Less code is required if only one compound subroutine needs to be called; It improves understanding of code; **<u>Data composition</u>** Array elements are easier to address than individual variables; The grouped data items/record can be manipulated as one unit; **Max 2** Award marks for **either** procedural composition **or** data composition **or** both.	**2**