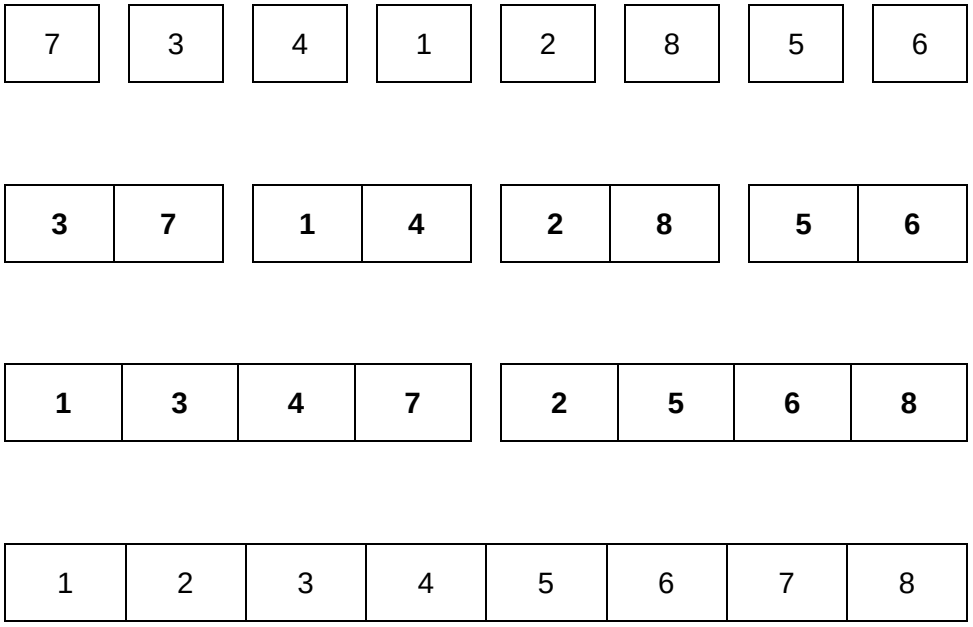
Question	Part	Marking guidance	Total marks
01	1	**Mark is for AO2 (apply)** Boolean//bool; **I.** Minor spelling mistakes	1
01	2	**2 marks for AO2 (apply)** (The identifier) `sorted` describes the purpose//role//meaning of the variable; this makes the algorithm easier to understand//maintain//follow; or (The identifier) `s` does not describe the purpose//role//meaning of the variable; this makes the algorithm harder to understand//maintain//follow;	2
01	3	**Mark is for AO2 (apply)** **A** (The algorithm uses a named constant.) only; **If more than one lozenge shaded then mark is not awarded**	1
01	4	**6 marks for AO2 (apply)** 1 mark for column `arr[0]` correct; 1 mark for column `arr[1]` correct; 1 mark for column `arr[2]` correct **only if** `arr[0]` and `arr[1]` are correct; 1 mark for `sorted` column correct; 1 mark for `i` column correct; 1 mark for `t` column correct; **I.** different rows used as long as the order within columns is clear **I.** duplicate values on consecutive rows within a column	6

Arr			sorted	i	t
0	1	2			
4	1	6	false		
			true	0	
1	4		false		4
				1	
				2	
			true	0	
				1	
				2	

Question	Part	Marking guidance	Total marks
01	5	**3 marks for AO2 (apply)** 1 mark if pairwise comparisons are made in the second row but allow for one pairwise comparison error; 1 mark if pairwise comparisons are made in the third row but allow for one pairwise comparison error (allow follow through from previous row); 1 mark if all correct; 	3
01	6	**Mark is for AO1 (understanding)** It is more (time) efficient// It will usually take fewer steps; **A.** quicker//it will take less time as long as the answer has been qualified.	1
01	7	**2 marks for AO2 (apply)** Maximum of 2 from: It allows the code to be (more easily) reused; It can be used to sort any array (not just the one on line 1); It would be easier to test; The code could be changed//updated without affecting the overall program; Makes the program easier to read//understand; **A.** Any other creditable answer as long as they are clearly distinct from the other responses.	2

Qu	Part	Marking guidance	Total marks
02		**4 marks for AO2 (apply)** Maximum 4 marks from: **If bubble sort chosen then:** 8 & 4 are swapped; 1 & 8 are swapped; 5 & 8 are swapped; 1 & 4 are swapped; swap two consecutive numbers if the left number was greater than the right number; would repeat passes until no swaps are made/all numbers are sorted // a pass of the array `[1, 4, 5, 8]` requiring no swaps and so the algorithm stops; *or by diagram:* *8 4 1 5* ; (both lines required) 4 8 1 5 4 1 8 5 ; 4 1 5 8 ; 1 4 5 8 ; **R.** the final (sorted) array if no prior arrays (excluding `[8, 4, 1, 5]`) are given. **If merge sort chosen then:** separate the array into arrays that contain only one element;; combine pairs of arrays, ordering the numbers // the values 8 and 4 combine to form the array `[4, 8]` and the value 1 and 5 combine to form the array `[1, 5]`; the arrays `[4, 8]` and `[1, 5]` combine to form the array `[1, 4, 5, 8]` / sorted array // 4 is compared with 1, 4 is compared with 5, 8 is compared with 5; Or by diagram (to a max 4 marks): `8, 4, 1, 5` ; (both lines required) `8, 4  1, 5` `8  4  1  5` ; `4, 8  1, 5` ; `1, 4, 5, 8` ; [A] **R.** mark [A] if preceding row not given.	4

Question	Part	Marking guidance	Total marks
03	1	**6 marks for AO2 (apply)** 1 mark for `i` column and `j` column initialised to `0`; 1 mark for rest of `i` **and** `j` columns correct; 1 mark for `temp` column correct; 1 mark for first swap setting `arr[0]` column to `b` **and** `arr[1]` column to `c`; 1 mark for second swap setting `arr[1]` column to `a` **and** `arr[2]` column to `c`; 1 mark for third swap setting `arr[0]` column to `a` **and** `arr[1]` column to `b`;	6

arr			i	j	temp
[0]	[1]	[2]			
c	b	a			
			0	0	c
b	c			1	c
	a	c	1	0	b
a	b			1	

I. Different rows used as long as the order within columns is clear
I. Duplicate values on consecutive rows within a column
I. Quotes used around letters
I. Case

Note to Examiners: **A.** missing middle `c` in **`temp`** column

Question	Part	Marking guidance	Total marks
03	2	**Mark is for AO2 (apply)** Sort (the values in order) // bubble sort // put into alphabetical order;	1

Question	Part	Marking guidance	Total marks
03	3	**Mark is for AO2 (apply)** The algorithm will attempt to access an element/item/index in the array that does not exist; // The algorithm will attempt to use an index which is greater than the maximum array index of 2;	1

Question	Part	Marking guidance	Total marks
04		**4 marks for AO1 (understanding)** **Maximum of 4 marks** from: ● The list is (repeatedly) divided into sub-lists (half the size / at the midpoint) until each sub-list is of length 1 (singleton lists) // The list is divided recursively until each sub-list is of length 1 (singleton lists); ● The algorithm compares / sorts individual elements as pairs; ● (After comparing) the individual items are then (repeatedly) merged back together into sub-lists // (After comparing) the sub-lists are then (repeatedly) merged back together; ● (Finally,) one list is produced in the right order (which is the sorted list); **Note to Examiners:** It is perfectly acceptable for the response given to be based on the contents of Figure 7 rather than a generic explanation.	4

Question	Part	Marking guidance	Total marks
05		**5 marks for AO2 (apply)** **1 mark** for each correct change (allow follow on); The correct sequence is: 3 **1** **5** 4 2 3 1 **4** **5** 2 3 1 4 **2** **5** **1** **3** 4 2 5 1 3 **2** **4** 5	5