# Part A – General short questions –

Give short answer for the following:

1. what is a queue? An abstract data type that holds an ordered, linear sequence of items. It uses a First In, First Out (FIFO) structure
2. What is meant by “Popping from a queue”? popping in a queue is the same as deleting from the queue
3. What is meant by “Pushing into a queue”? pushing in a queue is the same as adding to the queue
4. Why is it important to test if the queue is empty before popping an element? If there is nothing in the queue how are you supposed to delete anything
5. why is it important to test if the queue is full before pushing an element? If the queue is full it
6. Write pseudo code to check if queue is empty.
7. write pseudo code to check is queue is full.
8. write algorithm to push in to a queue using an array.

 9. write algorithm to pop from queue using an array.

|  |
| --- |
| Part B – Past exams Questions |
| Answer all questions below   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | **2(a).** | A program stores a queue of mathematical questions to be asked to a user. The questions are asked in the order they are added. Once a question has been asked it cannot be asked again. New questions are continually added to the end of the queue.  The program will use a non-circular queue, questions, (implemented using an array) to store the questions. The pointer, head, stores the index of the first element in the queue. The pointer, tail, stores the index of the last element in the queue.  Complete the following algorithm, to ask the user to input a new question and then either add it to the queue, or report that the queue is full. procedure add()   |  |  | | --- | --- | |  | maxElements = 10 |                   endprocedure   |  |  | | --- | --- | |  | **[4]** | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | **(b).** | Describe why a queue is a suitable structure for this program.            **[3]** | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | **(c).** | Complete the following algorithm, to remove, and output, the first element in the queue. procedure remove()                  endprocedure   |  |  | | --- | --- | |  | **[4]** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | **(d).** | Fig. 4.1 shows an example of the data in the queue. head is currently 0, tail is currently 4.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | “2\*3” | “1+4” | “3–1” | “10/2” | “3+6” |  |  |  |  | | Fig. 4.1 | | | | | | | | | |  1. Show the contents of the queue shown in Fig. 4.1, after the following code is run.  |  |  | | --- | --- | |  | add("6+1") |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | |  | **[2]** |  1. State the values stored in head and tail after the code in **part (i)** has run.  |  |  | | --- | --- | | head |  | | tail |  |  |  |  | | --- | --- | |  | **[2]** | | | |

**END OF QUESTION paper**

**Check your answer with Mark scheme in the next page**

# Mark scheme

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Question** | | | **Answer/Indicative content** | **Marks** | **Guidance** |
| 2 | a |  | 1 mark for pseudocode / code that meets each bullet   * Input a question [1] * Check if tail is full and outputs message / reports error [1] * Increment tail [1] * Adds question to tail of questions [1]   e.g. C:\core\files\questions\1529414448\H046H446-CompSc-H046-02-Jun17\img\pg16_A_02_150.png | **4    AO3.2 (4)** | **Examiner's Comment:** Again, the use of pseudocode posed problems for many candidates. Those who had a wider programming experience were apparent from the well-crafted solutions. Those who gained credit generally gained two marks for understanding how the pointers were updated and how data was added / removed. Fewer scored full marks by also performing error checking. |
|  | b |  | 1 mark per bullet to max 3 e.g.   * A queue is First In First Out (FIFO) [1] * The questions are retrieved in the order they are stored [1] * Questions can be added to the end [1] * Dynamic structure… [1] * …expands to take more questions [1] | **3  AO1.2   (2) AO2.1 (1)** | **Examiner's Comment:** Many candidates understood that a queue was a FIFO structure, but fewer could then go on to explain in context why this would then be a suitable data structure for the problem in context. |
|  | c |  | 1 mark for pseudocode/code that meets each bullet   * Checking if queue is empty [1] * …outputting message/reporting error [1] * Outputting element in questions at index head [1] * Increment head [1]   e.g. C:\core\files\questions\adminupload\114890\pg16_A_01a_150.png | **4   AO3.2 (4)** | **Examiner's Comment:** Again, the use of pseudocode posed problems for many candidates. Those who had a wider programming experience were apparent from the well-crafted solutions. Those who gained credit generally gained two marks for understanding how the pointers were updated and how data was added/removed. Fewer scored full marks by also performing error checking. |
|  | d | i | "6+1" in the correct box. [1] Speech marks present [1]   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | "2\*3" | "1+4" | "3-1" | "10/2" | "3+6" | "6+1" |  |  | | **2 AO2.1  (2)** | **Examiner's Comment:** Many candidates would have scored well on this question if they understood that a queue is FIFO. Those who did not understand the basic properties of a queue struggled with the question. |
|  |  | ii | 1 mark for head, 1 for tail head = 0 [1] tail = 5 [1] | **2  AO2.1 (2)** | **Examiner's Comment:** Many candidates would have scored well on this question if they understood that a queue is FIFO. Those who did not understand the basic properties of a queue struggled with the question. |
|  |  |  | **Total** | **15** |  |