A Level · OCR · Computer Science 38 mins 11 questions

Exam Questions

4.2 Data Structures

Arrays / Records, Lists & Tuples / Linked Lists / Stacks / Queues / Graphs / Graphs: Traversing, Adding & Removing Data / Trees / Binary Search Trees / Hash Tables

Easy (3 questions)	/4
Medium (5 questions)	/20
Hard (3 questions)	/14
Total Marks	**/38**

Scan here to return to the course
or visit savemyexams.com

Easy Questions

1 A stack is one data structure that is available for Sundip to use. She could also use a queue, list, linked list, array or tuple.

State how a tuple is different to a list.

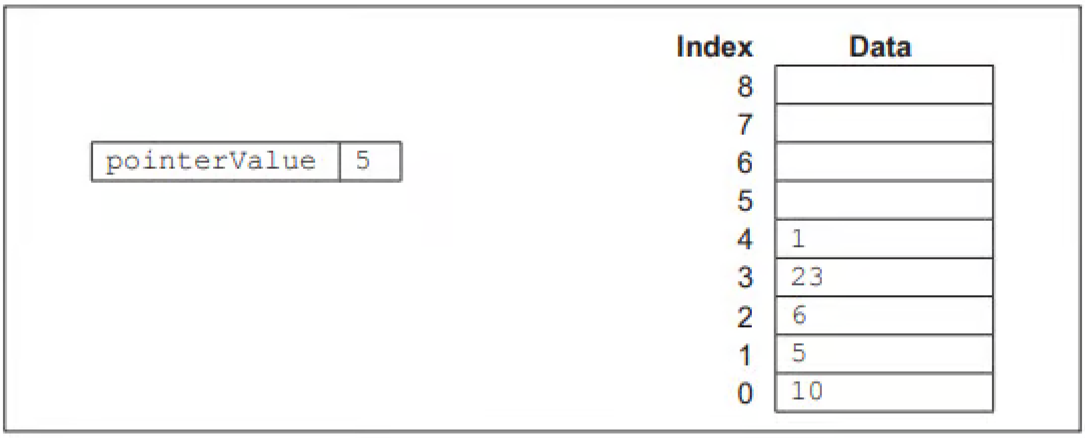
....

(1 mark)

2 A computer uses a stack data structure, implemented using an array, to store numbers entered by the user.

The array is zero based and has 100 locations.

Fig. 8 shows the current contents of the stack and the first 9 locations of the array



pointerValue	5

Index	Data
8	
7	
6	
5	
4	1
3	23
2	6
1	5
0	10

State the purpose of pointerValue

....

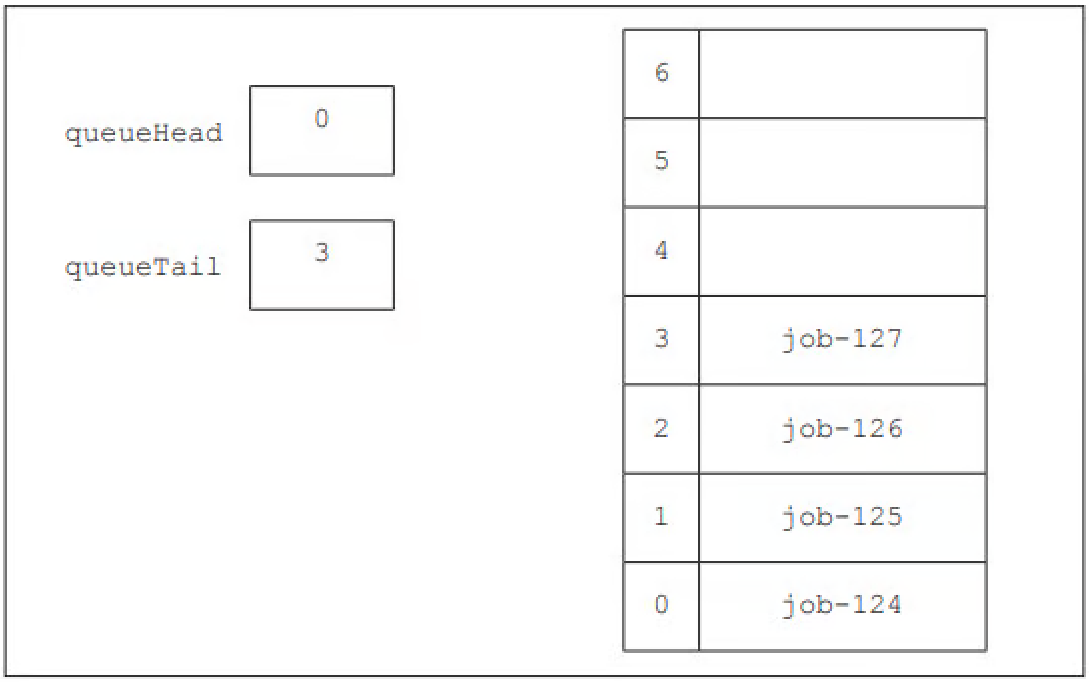
(1 mark)

3 A printer buffer is a storage area that holds the data, known as jobs, that are to be printed by a printer.

A simulation of the printer buffer uses a queue data structure to store jobs that are waiting to be printed. The queue is not circular.

The printer buffer is represented as a zero-indexed 1D array with the identifier buffer.

Fig. 2 shows the current contents of the queue buffer and its pointers.



6	
5	
4	
3	job-127
2	job-126
1	job-125
0	job-124

State the purpose of the pointers queueHead and queueTail.

....

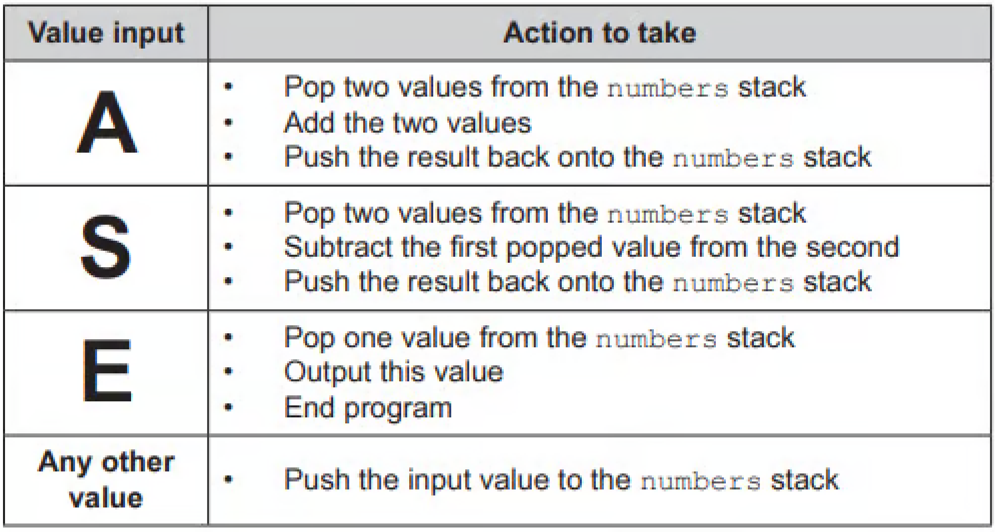
....

(2 marks)

Medium Questions

1 Sundip writes an algorithm to carry out addition and subtraction. The algorithm will use an initially empty stack with the identifier numbers and will take input from the user.

The action the algorithm takes depends on the value input by the user. These actions are listed in **Fig. 2**.



Value input	Action to take
A	• Pop two values from the `numbers` stack • Add the two values • Push the result back onto the `numbers` stack
S	• Pop two values from the `numbers` stack • Subtract the first popped value from the second • Push the result back onto the `numbers` stack
E	• Pop one value from the `numbers` stack • Output this value • End program
Any other value	• Push the input value to the `numbers` stack

A stack is one data structure that is available for Sundip to use. She could also use a queue, list, linked list, array or tuple.

Describe how the second item in a linked list would be accessed using pointer values.

....

....

....

(3 marks)

2 A computer uses a stack data structure, implemented using an array, to store numbers entered by the user.

The array is zero based and has 100 locations.

The program is amended to include the use of several queue data structures

Describe how an array can be used to implement a queue data structure.

(3 marks)

3 A business uses an array with the identifier wNames to store workers' names. A variable with the identifier top is used to store the index of the last element to be added to the array, which is also the element which will next be removed.

wNames

0	1	2	3	4	5	6
Kirstie	Martyn	Louise	Alex	Anna		

top

4

The same workers' names are stored in a **binary search tree** which is ordered alphabetically

Kirstie is set as the root node, with Martyn, Louise, Alex and Anna added one by one.

Complete the tree diagram above to show where Martyn, Louise, Alex and Anna would be added to this binary search tree.

....

....

....

....

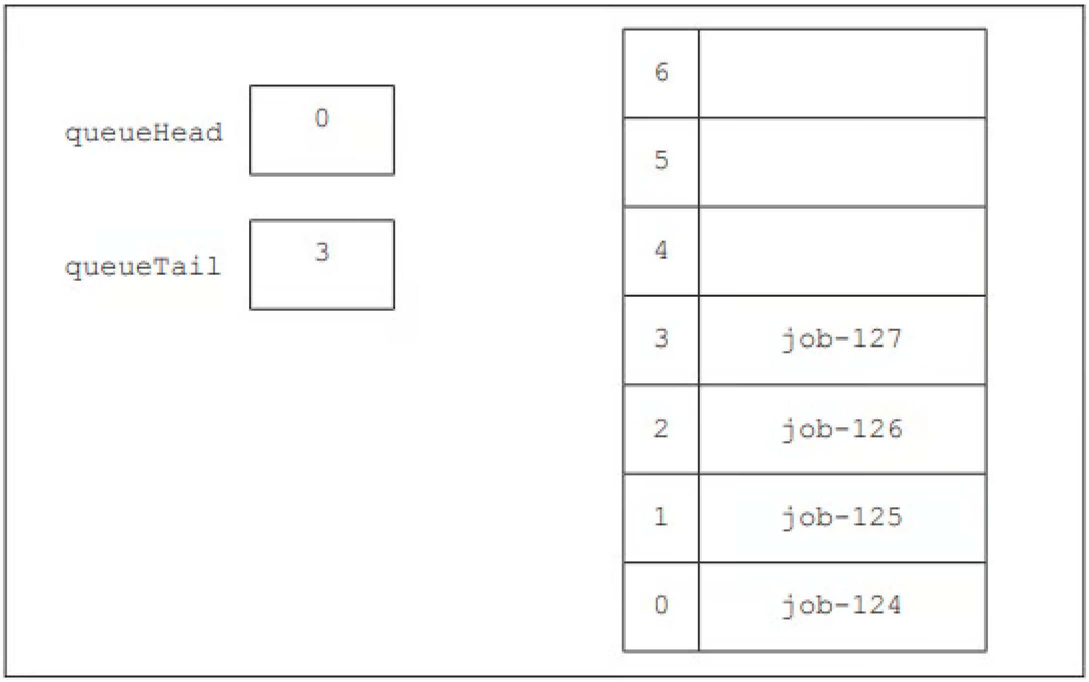
(4 marks)

4 A printer buffer is a storage area that holds the data, known as jobs, that are to be printed by a printer.

A simulation of the printer buffer uses a queue data structure to store jobs that are waiting to be printed. The queue is not circular.

The printer buffer is represented as a zero-indexed 1D array with the identifier buffer.

Fig. 2 shows the current contents of the queue buffer and its pointers.



queueHead	0
queueTail	3

6	
5	
4	
3	job-127
2	job-126
1	job-125
0	job-124

The array, buffer and pointer values are declared with global scope.

The function dequeue returns null if the array is empty, and the contents of the next element if not empty. The queue is not circular.

Write an algorithm, using pseudocode or program code, for the function dequeue().

....

....

....

....

....

(5 marks)

5 The function dequeue outputs and removes the next data item in the queue.

The procedure enqueue adds the job passed as a parameter to the queue.

Show the final contents of the queue and pointer values after the following instructions have been run on the queue buffer shown in **Fig. 2.**

```
dequeue()
dequeue()
enqueue(job-128)
```

dequeue()

enqueue(job-129)

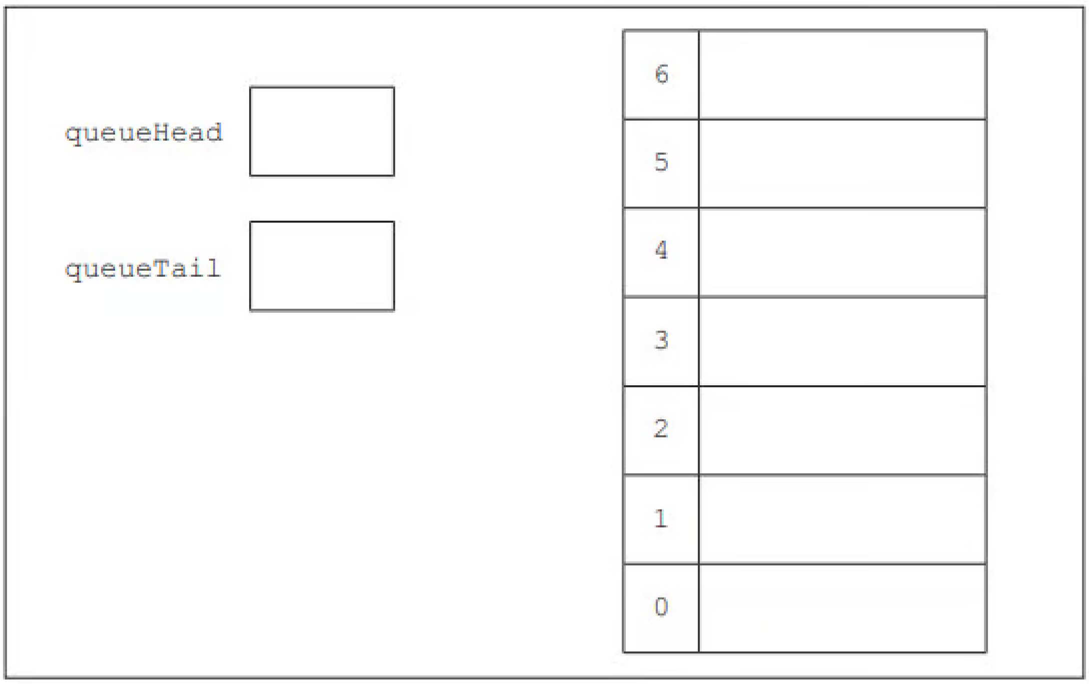


Fig.2

(5 marks)

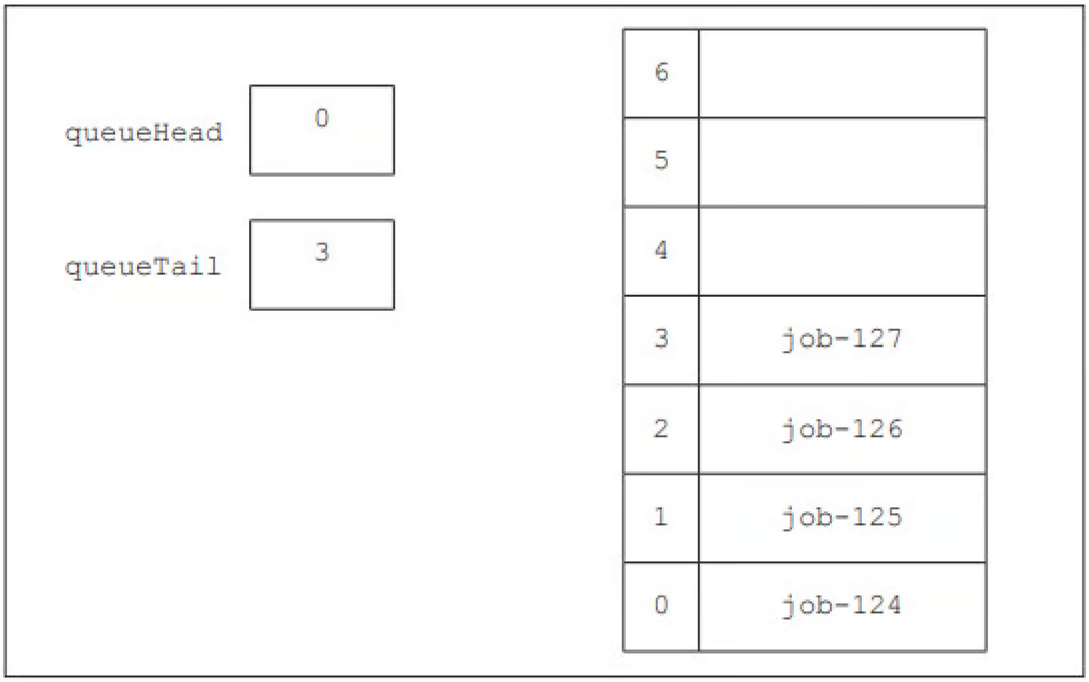
Hard Questions

1 A printer buffer is a storage area that holds the data, known as jobs, that are to be printed by a printer.

A simulation of the printer buffer uses a **queue data structure** to store jobs that are waiting to be printed. The queue is not circular.

The printer buffer is represented as a zero-indexed 1D array with the identifier `buffer`.

Fig. 2 shows the current contents of the queue buffer and its pointers.



6	
5	
4	
3	job-127
2	job-126
1	job-125
0	job-124

Some print jobs can have different priorities. The higher the priority the sooner the job needs to be printed.

Describe how the program could be changed to deal with different priorities.

....

....

....

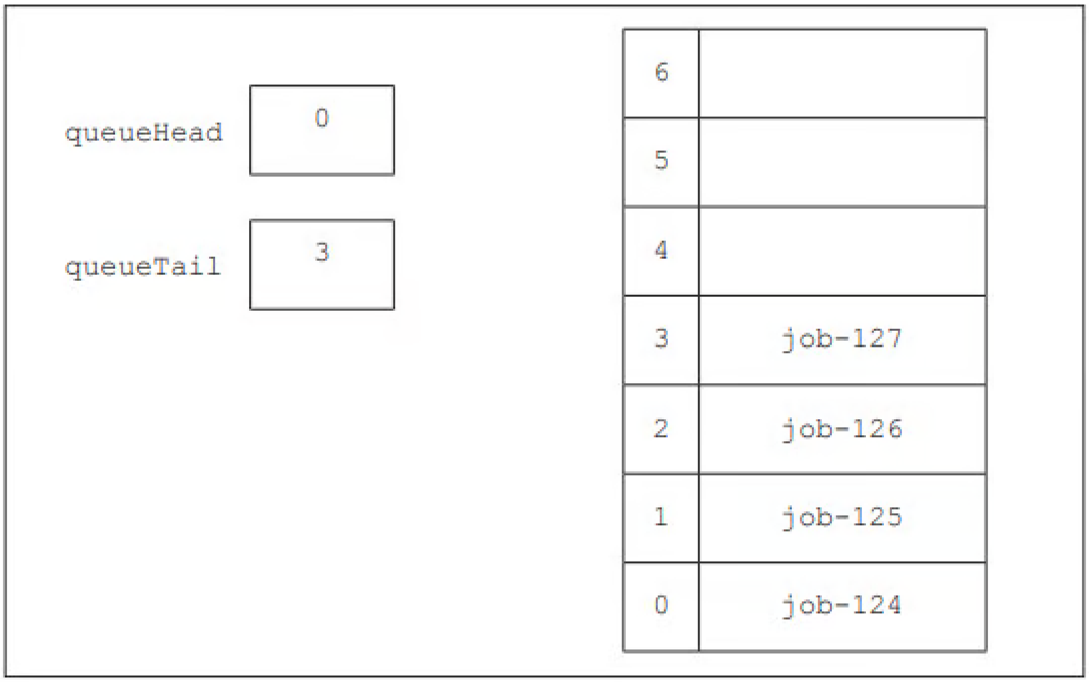
(3 marks)

2 A printer buffer is a storage area that holds the data, known as jobs, that are to be printed by a printer.

A simulation of the printer buffer uses a **queue data structure** to store jobs that are waiting to be printed. The queue is not circular.

The printer buffer is represented as a zero-indexed 1D array with the identifier buffer.

Fig. 2 shows the current contents of the queue buffer and its pointers.



queueHead 0

queueTail 3

6	
5	
4	
3	job-127
2	job-126
1	job-125
0	job-124

Some print jobs can have different priorities. The higher the priority the sooner the job needs to be printed.

The function dequeue returns null if the array is empty, and the contents of the next element if not empty. The queue is not circular.

The function enqueue returns -1 if there is no space at the end of the queue to add data, and returns 1 if the parameter was added to buffer. The array buffer contains a maximum of 100 elements.

The array, buffer and pointer values are declared with **global scope**

Write, using pseudocode or program code, an algorithm for the main program of the simulation. It should:

- In the main program of the simulation the user is asked whether they want to add an item to the queue or remove an item.
- If they choose to add an item they have to input the job name, and the function enqueue is called.
- If they choose to remove an item, the function dequeue is called and the job name is output.
- Appropriate messages are output if either action cannot be run because the queue is either empty or full.

(8 marks)

3 The procedure printLinkedList() follows the pointers to print all of the elements in the linked list.

```
01 procedure printLinkedList(headPointer)
02   tempPointer = headPointer - 1
03   dataToPrint = ""
04   if tempPointer == -1 then
05     print("List is full")
06   else
07     while linkedList[pointer].getPointer() != -1
08       dataToPrint = dataToPrint + " " + linkedList[tempPointer,0]
09       linkedList[tempPointer].getPointer() = tempPointer
10     endwhile
11   print(dataToPrint + " " + linkedList[tempPointer].getData()
12   endif
13 endprocedure
```

The procedure has a number of errors.

Identify three lines that contain an error and **write** the corrected line.

(3 marks)