IDMSQ01

– A stack S has 05 character items, S={“A”,“B”,“C”,“D”,“E”} where “E” is the top of S. What is the content of S if we perform the following list of operations on the stack: push(“F”) –-> pop() –-> pop() –-> pop() –-> push(“D”)?

Select one:

- S={“A”,“B”,“D”,“F”}
- S={“A”,“B”,“C”,“D”}
- S={“C”,“D”,“E”,“F”}
- S={“B”,“E”, “F” ,“D”}

IDMSQ02

A queue Q has 05 character items, Q={“A”, “B”, “C”, “D”, “E”} where “E” is the rear and “A” is the front of the queue. What is the content of Q if we perform the following list of operations on the queue: enqueue(“F”)-->dequeue()-->dequeue()-->dequeque()-->enqueue(“D”)?

Select one:

- Q={“C”, “D”, “E”, “F”}
- Q={“A”, “B”, “C”, “D”}
- Q={“D”, “F”, “A”, “B”}
- Q={“D”, “E” ,“F” ,“D”}

IDMSQ03

– A stack S has 05 character items, S={“5”, “4”, “3”, “2”, “1”} where “1” is the top of S. Which operations must be perform to change S into a new state: S={“5”, “4”, “2”, “3”, “1”}?

Select one:

- pop()-->pop()-->pop()-->push(“2”)-->push(“3”)-->push(“1”)
- pop()-->push(“2”)-->pop()-->push(“3”)-->pop()-->push(“1”)
- push(“2”)-->pop()-->push(“3”)-->pop()-->push(“1”)-->pop()
- push(“2”)-->push(“3”)-->push(“1”)-->pop()-->pop()-->pop()

IDMSQ04

– A queue Q has 05 character items, Q={“5”, “4”, “3”, “2”, “1”} where “1” is the front and “5” is the rear of Q. Which operations must be perform to change Q into a new state: Q={“3”, “2”, “1”, “4”, “5”}?

Select one:

- enqueue(“4”)-->enqueue(“5”)-->dequeue()-->dequeue()
- dequeue()-->enqueue(“5”)-->dequeue()-->enqueue(“4”)
- enqueue(“5”)-->enqueue(“4”)-->dequeue()-->dequeue()
- dequeue()-->dequeue()-->enqueue(“5”)-->enqueue(“4”)

IDMSQ05

In method F below, the stack s contains character items. Which is the result if we call method F with the input string text=“datastructure”?

```
public static void F(String text)
{
  Stack s=new Stack();
  for (int i=0; i<text.length(); i++)
    s.push(text.charAt(i));
  while (!s.isEmpty())
   System.out.print(s.pop());
}
```

Select one:

- erutcurtsatad
- datastructure
- erutcurtsataderutcurtsatad
- datastructuredatastructure

IDMSQ07

– One difference between a queue and a stack is:

Select one:

- Stacks use two ends of the structure, queues use only one.
- Queues require linked lists, but stacks do not.
- Queues use two ends of the structure; stacks use only one.
- Stacks require linked lists, but queues do not.

IDMSQ08

– In an array-based stack, which operation has time complexity O(N) in the worst-case?

Select one:

- push().

No operation that has time complexity O(N).

isEmpty().

pop().

IDMSQ09

– In an array-based circular queue, which operation has time complexity O(N) in the worst-case?

Select one:

isFull().

enqueue().

No operation that has time complexity O(N).

dequeue().

IDMSQ10

– Suppose that you are implementing an operation named multiDequeue(int k) on a queue contains integer items. This operation will perform dequeue() k times and return the result of the kth dequeue(). Please complete the code of the operation?

```
public int multiDequeue(int k)
{
    int m=k;
    int result;
    while ((!isEmpty())&&(m>0))
    {
       result=dequeue();
       ----;
    }
    return result;
}
```

Select one:

k=k-1

dequeue(k)

enqueue(k)

m=m-1

IDMSQ11

– Suppose that you are implementing an operation name multiPop(int k) on a stack contains integer items. This operation will perform pop() k times and return the result of the kth pop().

Please complete the code of the operation?

```
public int multiPop(int k)
{
    int m=k;
    int result;
    while ((!isEmpty())&&(m>0))
    {
       result=pop();
       ----;
    }
    return result;
}
```

Select one:

- k=k-1
- pop(k)
- m=m-1
- push(k)

IDMSQ12

A queue q has 5 items. How many items left in q after executing: q.enqueue(q.dequeue())?

Select one:

- 0
- 4
- 6
- 5

IDMSQ13

– In Java the run-time error "Array index out of bound" is reported when a program try to access an array with an index that is outside the valid boundaries of the array. Which of the following data structure may give the "Array index out of bound" error, even though the current number of elements in it, is less than its size.

Select one:

- Array-based stack.
- Circular array-based queue.
- Simple array-based queue.
- Array-based list.