Question 1

Correct

Mark 1.00 out of 1.00

Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
{
   IntQueue q = new IntQueue();
   q.enqueue(0);
   q.enqueue(1);
   for (int i = 0; i < n; i++)
   {
      int a = q.dequeue();
      int b = q.dequeue();
      q.enqueue(b);
      q.enqueue(a + b);
      ptint(a);
   }
}
```

Select one:

- a. Prints numbers from 0 to n-1
- b. Prints first n Even numbers.
- c. Prints first n Fibonacci numbers ✓

**Explanation:** The function prints first n Fibonacci Numbers. Note that 0 and 1 are initially there in q. In every iteration of loop sum of the two queue items is enqueued and the front item is dequeued.

- d. Prints numbers from n-1 to 0

The correct answer is: Prints first n Fibonacci numbers

Question 2

Correct

Mark 1.00 out of 1.00

Let Q denote a queue containing sixteen numbers and S be an empty stack. Head(Q) returns the element at the head of the queue Q **without** removing it from Q. Similarly Top(S) returns the element at the top of S **without** removing it from S. Consider the algorithm given below.

```
while Q is not Empty do
    if S is Empty OR Top(S) ≤ Head(Q) then
        x := Dequeue(Q);
        Push(S,x);
    else
        x := Pop(S);
        Enqueue(Q,x);
    end
end
```

The maximum possible number of iterations of the while loop in the algorithm is_____

Select one:

- a. 32
- b. 64
- c. 256 ✓

**Explanation:** The worst case happens when the queue is sorted in decreasing order. In worst case, loop runs n*n times.

- d. 16

The correct answer is: 256

**Question 3**

Correct

Mark 1.00 out of 1.00

What does the following function do for a given Linked List with first node as *head*?

```
void fun1(struct node* head)
{
  if(head == NULL)
    return;

  fun1(head->next);
  printf("%d  ", head->data);
}
```

Select one:

- a. Prints alternate nodes of Linked List
- b. Prints alternate nodes in reverse order
- c. Prints all nodes of linked lists
- d. Prints all nodes of linked list in reverse order ✓

**Explanation:** fun1() prints the given Linked List in reverse manner. For Linked List 1->2->3->4->5, fun1() prints 5->4->3->2->1.

The correct answer is: Prints all nodes of linked list in reverse order

**Question 4**

Correct

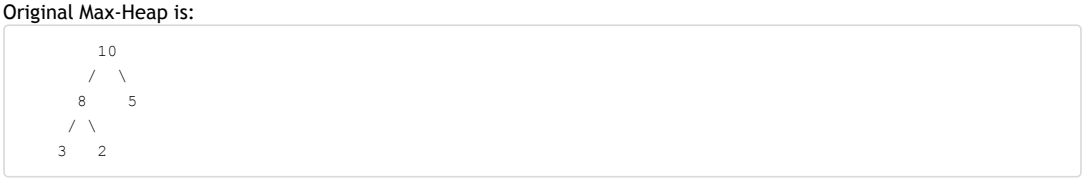
Mark 1.00 out of 1.00

A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below:
10, 8, 5, 3, 2
Two new elements "1" and "7" are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

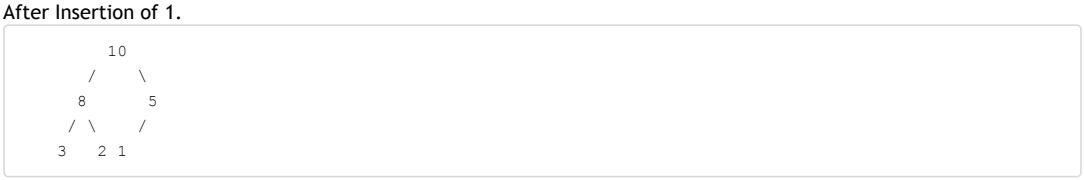
Select one:

- a. 10, 8, 7, 2, 3, 1, 5
- b. 10, 8, 7, 5, 3, 2, 1
- c. 10, 8, 7, 3, 2, 1, 5 ✓

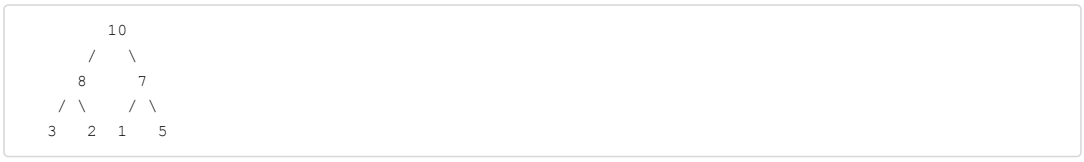
**Explanation:**
Original Max-Heap is:




After Insertion of 1.




After Insertion of 7.




- d. 10, 8, 7, 1, 2, 3, 5

The correct answer is: 10, 8, 7, 3, 2, 1, 5

**Question 5**

Correct

Mark 1.00 out of 1.00

An implementation of a queue Q, using two stacks S1 and S2, is given below:

```
void insert(Q, x) {
   push (S1, x);
}

void delete(Q){
   if(stack-empty(S2)) then
      if(stack-empty(S1)) then {
         print("Q is empty");
         return;
      }
      else while (!(stack-empty(S1))){
         x=pop(S1);
         push(S2,x);
      }
   x=pop(S2);
}
```

Let n insert and m (<=n) delete operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n?

Select one:

- a. 2m <= x < 2n and 2m <= y <= n+m
- b. 2m <= x <2n and 2m <= y <= 2n
- c. n+m <= x < 2n and 2m<= y <= 2n
- d. n+m <= x < 2n and 2m <= y <= n+m ✓

**Explanation:** The order in which insert and delete operations are performed matters here.
The best case: Insert and delete operations are performed alternatively. In every delete operation, 2 pop and 1 push operations are performed. So, total m+ n push (n push for insert() and m push for delete()) operations and 2m pop operations are performed.
The worst case: First n elements are inserted and then m elements are deleted. In first delete operation, n + 1 pop operations and n push operation are performed. Other than first, in all delete operations, 1 pop operation is performed. So, total m + n pop operations and 2n push operations are performed (n push for insert() and n push for delete())

The correct answer is: n+m <= x < 2n and 2m <= y <= n+m

**Question 6**

Not answered

Marked out of 1.00

What is the output of the following code snippet?

```
3: int x1 = 50, x2 = 75;
4: boolean b = x1 >= x2;
5: if(b = true) System.out.println("Success");
6: else System.out.println("Failure");
```

Select one or more:

- a. Failure
- b. The code will not compile because of line 5.
- c. The code will not compile because of line 4.
- d. Success

Your answer is incorrect.

The correct answer is: Success

**Question 7**

Correct

Mark 1.00 out of 1.00

Which of the following compile? (Choose all that apply)

Select one or more:

- [ ] a. public void moreD(String... values, int... nums) {}
- [x] b. public void moreG(String[] values, int[] nums) {}

Options A and B are correct because the single vararg parameter is the last parameter declared.
Option G is correct because it doesn't use any vararg parameters at all.
Options C and F are incorrect because the vararg parameter is not last.
Option D is incorrect because two vararg parameters are not allowed in the same method.
Option E is incorrect because the ... for a vararg must be after the type, not before it

- [ ] c. public void moreF(String... values, int[] nums) {}
- [ ] d. public void moreC(int... nums, String values) {}
- [x] e. public void moreA(int... nums) {}
- [x] f. public void moreB(String values, int... nums) {}
- [ ] g. public void moreE(String[] values, ...int nums) {}

Your answer is correct.

The correct answer is: public void moreA(int... nums) {}, public void moreB(String values, int... nums) {}, public void moreG(String[] values, int[] nums) {}

**Question 8**

Correct

Mark 1.00 out of 1.00

Which are true of the following code? (Choose all that apply)

```
1: public class Rope {
2: public static void swing() {
3: System.out.print("swing ");
4: }
5: public void climb() {
6: System.out.println("climb ");
7: }
8: public static void play() {
9: swing();
10: climb();
11: }
12: public static void main(String[] args) {
13: Rope rope = new Rope();
14: rope.play();
15: Rope rope2 = null;
16: rope2.play();
17: }
18: }
```

Select one or more:

- [ ] a. There are exactly two compiler errors in the code.
- [x] b. There is exactly one compiler error in the code.
- [x] c. If the lines with compiler errors are removed, the output is swing swing.

Line 10 does not compile because static methods are not allowed to call instance methods.
Even though we are calling play() as if it were an instance method and an instance exists,
Java knows play() is really a static method and treats it as such. If line 10 is removed, the code works.
It does not throw a NullPointerException on line 16 because play() is a static method.
Java looks at the type of the reference for rope2 and translates the call to Rope.play().

- [ ] d. If the lines with compile errors are removed, the code throws a NullPointerException
- [ ] e. If the lines with compiler errors are removed, the output is climb climb
- [ ] f. The code compiles as is.

Your answer is correct.

The correct answer is: There is exactly one compiler error in the code., If the lines with compiler errors are removed, the output is swing swing.

**Question 9**

Correct

Mark 1.00 out of 1.00

What is the result of the following statements?

```
1: public class Test {
2: public void print(byte x) {
3: System.out.print("byte");
4: }
5: public void print(int x) {
6: System.out.print("int");
7: }
8: public void print(float x) {
9: System.out.print("float");
10: }
11: public void print(Object x) {
12: System.out.print("Object");
13: }
14: public static void main(String[] args) {
15: Test t = new Test();
16: short s = 123;
17: t.print(s);
18: t.print(true);
19: t.print(6.789);
20: }
21: }
```

Select one or more:

- [x] a. intObjectObject

The argument on line 17 is a short. It can be promoted to an int, so print() on line 5 is invoked.
The argument on line 18 is a boolean. It can be autoboxed to a bool-ean, so print() on line 11 is invoked.
The argument on line 19 is a double. It can be autoboxed to a double, so print() on line 11 is invoked.
Therefore, the output is intObjectObject and the correct answer is option E

- [ ] b. intObjectfloat
- [ ] c. byteObjectfloat
- [ ] d. bytefloatObject
- [ ] e. intfloatObject
- [ ] f. byteObjectObject

Your answer is correct.

The correct answer is: intObjectObject

**Question 10**

Correct

Mark 1.00 out of 1.00

Which code can be inserted to have the code print 2?

```
public class BirdSeed {
private int numberBags;
boolean call;
public BirdSeed() {
// LINE 1
call = false;
// LINE 2
}
public BirdSeed(int numberBags) {
this.numberBags = numberBags;
}
public static void main(String[] args) {
BirdSeed seed = new BirdSeed();
System.out.println(seed.numberBags);
} }
```

Select one or more:

- [ ] a. Replace line 1 with BirdSeed(2);
- [ ] b. Replace line 2 with new BirdSeed(2);
- [x] c. Replace line 1 with this(2);

Options A and B will not compile because constructors cannot be called without new.
Options C and D will compile but will create a new object rather than setting the fields in this one.
Option F will not compile because this() must be the first line of a constructor.
Option E is correct.

- [ ] d. Replace line 2 with this(2);
- [ ] e. Replace line 1 with new BirdSeed(2);
- [ ] f. Replace line 2 with BirdSeed(2);

Your answer is correct.

The correct answer is: Replace line 1 with this(2);