1) For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?

a) v=e

b) v = e+1

c) v + 1 = e

d) v = e-1

Answer: b

2) What is the number of edges present in a complete graph having n vertices?

a) (n\*(n+1))/2

b) (n\*(n-1))/2

c) n

d) Information given is insufficient

Answer: b

Explanation: Number of ways in which every vertex can be connected to each other is nC2.

3) Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree.

The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?

(A) 7 5 1 0 3 2 4 6 8 9

(B) 0 2 4 3 1 6 5 9 8 7

(C) 0 1 2 3 4 5 6 7 8 9

(D) 9 8 6 4 2 3 0 1 5 7

Answer: (C)

4)Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

void fun(Queue \*Q)

{

Stack S; // Say it creates an empty stack S

// Run while Q is not empty

while (!isEmpty(Q))

{

// deQueue an item from Q and push the dequeued item to S

push(&S, deQueue(Q));

}

// Run while Stack S is not empty

while (!isEmpty(&S))

{

// Pop an item from S and enqueue the popped item to Q

enQueue(Q, pop(&S));

}

}

What does the above function do in general?

(A) Removes the last from Q

(B) Keeps the Q same as it was before the call

(C) Makes Q empty

(D) Reverses the Q

Answer: (D)

Explanation: The function takes a queue Q as an argument. It dequeues all items of Q and pushes them to a stack S.

Then pops all items of S and enqueues the items back to Q.

Since stack is LIFO order, all items of queue are reversed.

5) Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?

(A) Possible if X is not last node. Use following two steps (a) Copy the data of next of X to X. (b) Delete next of X.

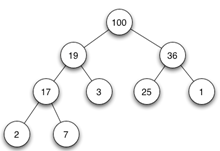
(B) Possible if size of linked list is even.

(C) Possible if size of linked list is odd

(D) Possible if X is not first node. Use following two steps (a) Copy the data of next of X to X. (b) Delete next of X.

Answer: (A)

6)



If we implement heap as maximum heap , adding a new node of value 35,. What values will be at leaf nodes of the left subtree of the heap.

A) 35 , 2 and 7

B) 3 and 2 and 7

C) 35 and 7

D) 2 and 7

Answer: B

Explanation: As 35 is greater than 3, so there is a violation and the node will swap at that position. So leaf nodes with value s are 2 , 7 and 3.

7)

import java.util.\*;

class Demo {

public static void main(String[] args)

{

ArrayList<Integer> arr = new ArrayList<Integer>();

arr.add(11);

arr.add(2);

arr.add(3);

arr.add(5);

arr.add(7);

arr.remove(new Integer(7));

arr.remove(2);

for (int i = 0; i < arr.size(); i++)

System.out.print(arr.get(i) + " ");

}

}

A. Compilation error.

B. 11 3 5

C. 11 2 5

D. Run time exception

Answer: C

8)

import java.util.\*;

class Demo {

public static void main(String[] args)

{

Deque<Integer> dq = new LinkedList<Integer>();

dq.offerFirst(1);

dq.offerFirst(2);

dq.offerFirst(3);

dq.offerLast(4);

Queue<Integer> q = new LinkedList<Integer>();

Iterator it = dq.descendingIterator();

while (it.hasNext()) {

System.out.print(it.next() + " ");

}

}

}

A. 1 2 3 4

B. 4 1 2 3

C. 4 3 2 1

D. Compilation fails

E. Run time exception

Answer: B.

9) What will be the output of the program?

class MyThread extends Thread

{

MyThread()

{

System.out.print(" MyThread");

}

public void run()

{

System.out.print(" bar");

}

public void run(String s)

{

System.out.println(" baz");

}

}

public class TestThreads

{

public static void main (String [] args)

{

Thread t = new MyThread()

{

public void run()

{

System.out.println(" foo");

}

};

t.start();

}

}

A. foo

B. MyThread foo

C. MyThread bar

D. foo bar

Answer: Option B

Explanation: Option B is correct because in the first line of main we're constructing an instance of an anonymous inner class extending from MyThread.

So the MyThread constructor runs and prints "MyThread".

The next statement in main invokes start() on the new thread instance, which causes the overridden run() method

(the run() method defined in the anonymous inner class) to be invoked, which prints "foo"

10) If the address of the first node of a linked list consisting of an integer and a

pointer to next node is 1000, what will be the address of the second node

of the same linked list ?

a)1004

b)1008

c)1006

d)1002

e)None of these

Ans- None of these