Day-33 MCQ

Question 1

Correct

Mark 1.00 out of 1.00

How will you find the minimum element in a binary search tree?

A.
```
public void min(Tree root)
{
    while(root.left() != null)
    {
        root = root.left();
    }
    System.out.println(root.data());
}
```

B.
```
public void min(Tree root)
{
    while(root != null)
    {
        root = root.left();
    }
    System.out.println(root.data());
}
```

C.
```
public void min(Tree root)
{   while(root.right() != null)
    {
        root = root.right();
    }
    System.out.println(root.data());
}
```

D.
```
public void min(Tree root)
{
    while(root != null)
    {
        root = root.right();
    }
    System.out.println(root.data());
```

Select one or more:

- [x] a. A

Explanation: Since all the elements lesser than a given node will be towards the left, iterating to the leftmost leaf of the root will give the minimum element in a binary search tree.

- [ ] b. B
- [ ] c. C
- [ ] d. D

Your answer is correct.

The correct answer is: A

Question 2

Correct

Mark 1.00 out of 1.00

The following lines talks about deleting a node in a binary tree.(the tree property must not be violated after deletion) i) from root search for the node to be deleted ii)_____ iii) delete the node at _____ what must be statement ii) and fill up statement iii)

A. ii)-find random node,replace with node to be deleted. iii)- delete the node

B. ii)-find node to be deleted. iii)- delete the node at found location

C. ii)-find deepest node,replace with node to be deleted. iii)- delete a node

D. ii)-find deepest node,replace with node to be deleted. iii)- delete the deepest node

Select one or more:

- [ ] a. A
- [ ] b. B
- [ ] c. C
- [x] d. D

Explanation : We just replace a to be deleted node with last leaf node of a tree. this must not be done in case of BST or heaps.

Your answer is correct.

The correct answer is: D

Question 3

Correct

Mark 1.00 out of 1.00

Select the code snippet that performs post-order traversal iteratively.

Select one or more:

a.

```
public void postorder(Tree root)
{
      if (root == null)
      return;
      Stack stk = new Stack();
      Tree node = root;
      while (!stk.isEmpty() || node != null)
      {
         while (node != null)
         {
            if (node.right != null)
               stk.add(node.left);
            stk.add(node);
            node = node.right;
         }
         node = stk.pop();
         if (node.right != null &&
!stk.isEmpty() && node.right == stk.peek())
         {
            Trees nodeRight = stk.pop();
            stk.push(node);
            node = nodeRight;
         } else
         {
            System.out.print(node.data + "
");
            node = null;
         }
      }
}
```

b.

```
public void postorder(Tree root)
{
      if (root == null)
      return;
      Stack stk = new Stack();
      Tree node = root;
      while (!stk.isEmpty() || node != null)
      {
         while (node != null)
         {
            if (node.right != null)
               stk.add(node.right);
            stk.add(node);
            node = node.left;
         }
         node = stk.pop();
         if (node.right != null &&
!stk.isEmpty() && node.right == stk.peek())
         {
            Trees nodeRight = stk.pop();
            stk.push(node);
            node = nodeRight;
         } else
         {
            System.out.print(node.data + "
");
            node = null;
         }      }}
```

✓

The left and right children are added first to the stack, followed by the node, which is then popped. Post-order follows LRN policy.

c.

```
public void postorder(Tree root)
{
      if (root == null)
         return;
      Stack stk = new Stack();
      Tree node = root;
      while (!stk.isEmpty() || node != null)
      {
         while (node != null)
         {
            if (node.right != null)
               stk.add(node.right);
            stk.add(node);
            node = node.left;
         }
         node = stk.pop();
         if (node.right != null)
         {
            Trees nodeRight = stk.pop();
            stk.push(node);
            node = nodeRight;
         } else
         {
            System.out.print(node.data + "
");
            node = null;
         }
      }
}
```

d. None of the mentioned

Your answer is correct.

The correct answer is:

```
public void postorder(Tree root)
{
      if (root == null)
      return;
      Stack stk = new Stack();
      Tree node = root;
      while (!stk.isEmpty() || node != null)
      {
         while (node != null)
         {
            if (node.right != null)
               stk.add(node.right);
            stk.add(node);
            node = node.left;
         }
         node = stk.pop();
         if (node.right != null &&
!stk.isEmpty() && node.right == stk.peek())
         {
            Trees nodeRight = stk.pop();
            stk.push(node);
            node = nodeRight;
         } else
         {
            System.out.print(node.data + "
");
            node = null;
         }      }}
```

Question 4

Correct

Mark 1.00 out of 1.00

What is the number of moves required in the Tower of Hanoi problem for k disks?

Select one or more:

- a. 2k - 1
- b. 2k + 1
- c. 2^k + 1
- d. 2^k - 1 ✓

Explanation : With 3 disks, the puzzle can be solved in 7 moves. The **minimal** number of moves **required** to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks.

Your answer is correct.

The correct answer is: 2^k - 1

Question 5

Correct

Mark 1.00 out of 1.00

Choose the code snippet which inserts a node to the head of the list?

Select one or more:

a.

```
public void insertHead(int data)
{
      Node temp = new Node(data);
      Node cur = head;
      while(cur.getNext() != head)
            cur = cur.getNext()
      if(head == null)
      {
            head = temp;
            head.setNext(head);
      }
      else
      {
            temp.setNext(head);
            head = temp;
            cur.setNext(temp);
      }
      size++;
}
```

✓

Explanation:
If the list is empty make the new node as 'head', otherwise traverse the list to the end and make its 'next' pointer point to the new node, set the new node's next point to the current head and make the new node as the head.

b.

```
public void insertHead(int data)
{
      Node temp = new Node(data);
      while(cur != head)
            cur = cur.getNext()
      if(head == null)
      {
            head = temp;
            head.setNext(head);
      }

      else
      {
            temp.setNext(head.getNext());
            cur.setNext(temp);
      }
      size++;
}
```

c.

```
public void insertHead(int data)
{
      Node temp = new Node(data);
      if(head == null)
      {
            head = temp;
            head.setNext(head);
      }
      else
      {
            temp.setNext(head.getNext());
            head = temp;
      }
      size++;
}
```

d.

```
public void insertHead(int data)
{
      Node temp = new Node(data);
      if(head == null)
      {
            head = temp;
            head.setNext(head.getNext());
      }
      else
      {
            temp.setNext(head.getNext());
            head = temp;
      }
      size++;
}
```

Your answer is correct.

The correct answer is:

```
public void insertHead(int data)
{
      Node temp = new Node(data);
      Node cur = head;
      while(cur.getNext() != head)
            cur = cur.getNext()
      if(head == null)
      {
            head = temp;
            head.setNext(head);
      }
      else
      {
            temp.setNext(head);
            head = temp;
            cur.setNext(temp);
      }
      size++;
}
```

Question 6

Correct

Mark 1.00 out of 1.00

What will be the output of the code snippet given below?

```
public class Main
{
public static void main (String [ ] args )
{
boolean a = true, b = false, c = true, d = ! ( a && b | | c ), e = ( d && a | | b ) && c, f = !d | | e ^ a && c ;
System.outprint ( d + " " + e + " " + f ) ;
}
}
```

Select one or more:

- [ ] a. false true false
- [ ] b. true true false
- [x] c. false flase true
- [ ] d. true false true

Your answer is correct.

The correct answer is: false flase true

Question 7

Correct

Mark 1.00 out of 1.00

What will be the output of the code snippet given below?

```
public class Main
{
public static void main (String [ ] args )
{
int x = 34, y = 23 ;
int z = x++ | ++y ;
int w = x++ | ++y ^ z++ ;
System.out.print( w + “ “ + x + “ “ + y + “ “ + z ) ;
}
}
```

Select one or more:

- a. 34 36 25 56
- b. 36 36 25 57
- c. 36 36 25 58
- d. 35 36 25 59 ✓

Your answer is correct.

The correct answer is: 35 36 25 59

Question 8

Correct

Mark 1.00 out of 1.00

Which of the following statement is correct about the code snippet given below?

```
public class Main
{
public static void main (String [ ] args )
{
float y = 2.304f ;
if ( y == 2.304 )
System.out.println ( “x and y are equal“ ) ;
else
System.out.println ( “x and y are not equal“ ) ;
}
}
```

Select one or more:

- a. The code reports an error
- b. The code displays an output as x and y are not equal. ✓

  Floating point values shall not be compared using either the == or != operators. Most floating point values have no exact binary representation and have a limited precision
- c. The code causes an exception.
- d. The code displays an output as x and y are equal.

Your answer is correct.

The correct answer is: The code displays an output as x and y are not equal.

Question 9

Correct

Mark 1.00 out of 1.00

What will be the output of the code snippet given below?

```
public class Main
{
public static void main (String [ ] args )
{
int i, j, k ;
for ( i = 0 ; i < 3 ; i++ )
{
for ( j = 0 ; j < 3 ; j++ )
{
for ( k = 0 ; k < 3 ; k++ )
{
If ( ( i == j ) && ( j == k ) )
{
System.out.print ( i + " " + j + " " k + " " ) ;
break ;
}
else
continue ;
}
}
}
}
}
```

Select one or more:

- [x] a. 0 0 0 1 1 1 2 2 2
- [ ] b. 0 1 2 1 0 2 1 2 0
- [ ] c. 0 1 2 1 2 0 2 0 1
- [ ] d. 0 1 2 0 1 2 0 1 2

Your answer is correct.

The correct answer is: 0 0 0 1 1 1 2 2 2

Question 10

Correct

Mark 1.00 out of 1.00

Which of the following statement is correct about the code snippet given below?

```
public class Main
{
public static void main (String [ ] args )
{
int a = 3, b = 5, c = 1 ;
if ( b == ( ++b && --c || ( a = !a ) ) )
System.out.print ( b + " " ) ;
if ( a == ( c || a-- ) )
System.out.print ( a + " " ) ;
if ( c != 1 )
System.out.print ( c + " " ) ;
}
}
```

Select one or more:

- [ ] a. The code displays an output as 1 1 .
- [ ] b. The code displays an output as 1 1 1 .
- [x] c. The code reports an error
- [ ] d. The code displays an output as 0 .

Your answer is correct.

The correct answer is: The code reports an error

Question 11

Correct

Mark 1.00 out of 1.00

Given the basic ER and relational models, which of the following is INCORRECT?

Select one or more:

- a. An attribute of an entity can be composite
- b. In a row of a relational table, an attribute can have more than one value ✓

A. An attribute of an entity can have more than one value
B. An attribute of an entity can be composite
C. In a row of a relational table, an attribute can have more than one value
D. In a row of a relational table, an attribute can have exactly one value or a NULL value
The term 'entity' belongs to ER model and the term 'relational table' belongs to relational model.
A and B both are true. ER model supports both multivalued and composite attributes
(C) is false and (D) is true.
In Relation model, an entry in relational table can can have exactly one value or a NULL.

- c. An attribute of an entity can have more than one value
- d. In a row of a relational table, an attribute can have exactly one value or a NULL value

Your answer is correct.

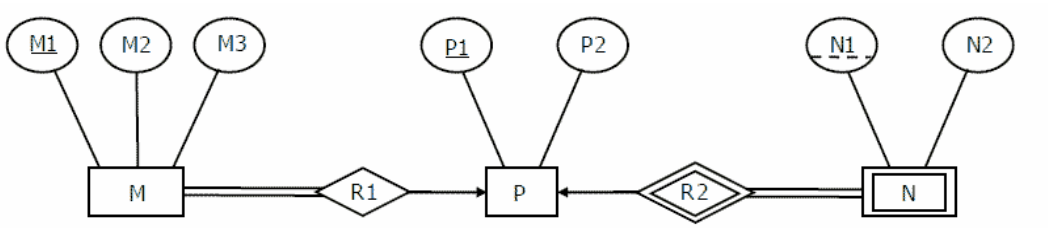
The correct answer is: In a row of a relational table, an attribute can have more than one value

Question 12

Correct

Mark 1.00 out of 1.00

Consider the following ER diagram.



The minimum number of tables needed to represent M, N, P, R1, R2 is ____.

Select one or more:

- a. 4
- b. 2
- c. 5
- d. 3 ✓

M, P are strong entities hence they must be represented by separate tables.
Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side.
(This way no extra table will be needed for Relationship sets) M table is modified to include primary key of P side(i.e. P1).
N is weak entity, and is modified to include primary key of P (i.e, P1).
Therefore there would be minimum of 3 tables with schema given below :
M (M1, M2, M3, P1)
P (P1, P2)
N (P1, N1, N2)

Your answer is correct.

The correct answer is: 3

Question 13

Correct

Mark 1.00 out of 1.00

Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes.
R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many and R2 is many-to-many.
R1 and R2 do not have any attributes of their own.
What is the minimum number of tables required to represent this situation in the relational model?

Select one or more:

- a. 5
- b. 4
- c. 2
- d. 3 ✓

Strong entities E1 and E2 are represented as separate tables. In addition to that many-to-many relationships(R2) must be converted as
seperate table by having primary keys of E1 and E2 as foreign keys.
One-to-many relaionship (R1) must be transferred to 'many' side table(i.e. E2) by having primary key of one side(E1)
as foreign key(this way we need not to make a seperate table for R1).
Let relation schema be E1(a1,a2) and E2(b1,b2). Relation E1(a1 is the key)
a1 a2
---
1 3
2 4
3 4
Relation E2(b1 is the key, a1 is the foreign key, hence R1(one-many) relationship set satisfy here)
b1 b2 a1
---
7 4 2
8 7 2
9 7 3
Relation R2 ({a1, b1} combined is the key here , representing many-many relationship R2)
a1 b1
---
1 7
1 8
2 9
3 9
Hence we will have minimum of 3 tables.

Your answer is correct.

The correct answer is: 3

Question 14

Correct

Mark 1.00 out of 1.00

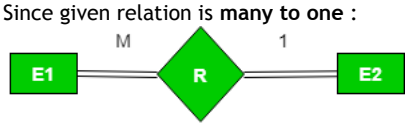
In an Entity-Relationship (ER) model, suppose R is a many-to-one relationship from entity set E1 to entity set E2.
Assume that E1 and E2 participate totally in R and that the cardinality of E1 is greater that the cardinality of E2.

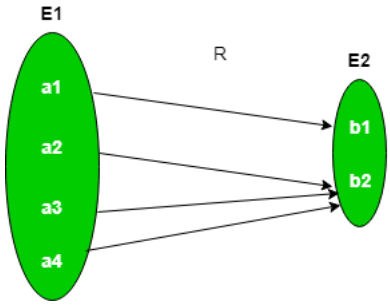
Which one of the following is true about R?

Select one or more:

- a. Every entity in E2 is associated with at most one entity in E1
- b. Every entity in E2 is associated with exactly one entity in E1.
- c. Some entity in E1 is associated with more than one entity in E2
- d. Every entity in E1 is associated with exactly one entity in E2. ✓

Since given relation is **many to one** :





Therefore, no entity in E1 can be related to more than one entity in E2 and an entity in E2 can be related to more than one entity in E1.
Only this option is correct.

Your answer is correct.

The correct answer is: Every entity in E1 is associated with exactly one entity in E2.

Question 15

Correct

Mark 1.00 out of 1.00

Integrity constraints ensure that changes made to the database by authorized users do not result into loss of data consistency.
Which of the following statement(s) is (are) true w.r.t. the examples of integrity constraints ?
(1) An instructor Id. No. cannot be null, provided Intructor Id No. being primary key.
(2) No two citizens have same Adhar-Id.
(3) Budget of a company must be zero.

Select one or more:

- a. (1), (2) and (3) are true.
- b. (1) and (2) are true , (3) false. ✓

(1) An instructor Id. No. cannot be null, provided Instructor Id No. being primary key. Correct by Codd's rule
(2) No two citizens have same Adhar-Id. Correct because Adhar is identification for citizens so it must be unique
(3) Budget of a company must be zero. We cant say or it is not necessarily true .
So, (1) and (2) are true , (3) false is correct.

- c. (1), (2) and (3) are false
- d. (1) false, (2) and (3) are true.

Your answer is correct.

The correct answer is: (1) and (2) are true , (3) false.

/