End Term Examination- 2020-2021

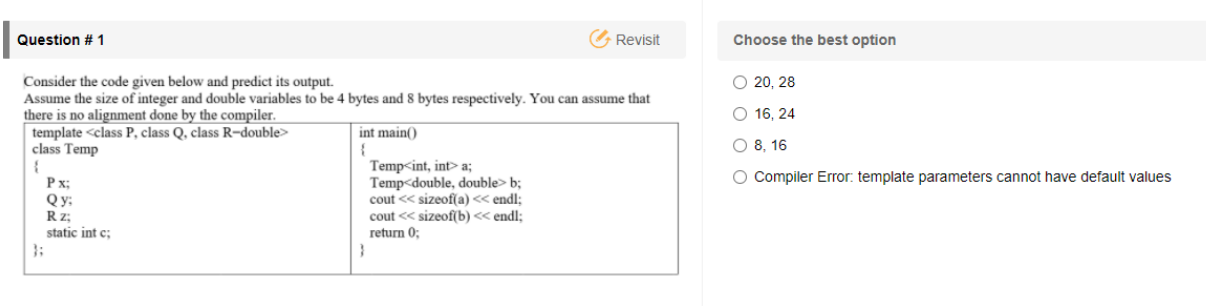
B.Tech., Odd Semester

Course Title: Data Structures

Course Code: 15B11CI311

Maximum Marks: 35

Maximum Time: 1.5 hr



Question # 1

Consider the code given below and predict its output.
Assume the size of integer and double variables to be 4 bytes and 8 bytes respectively. You can assume that there is no alignment done by the compiler.

```
template <class P, class Q, class R=double>
class Temp
{
   P x;
   Q y;
   R z;
   static int c;
};
```

```
int main()
{
   Temp<int, int> a;
   Temp<double, double> b;
   cout << sizeof(a) << endl;
   cout << sizeof(b) << endl;
   return 0;
}
```

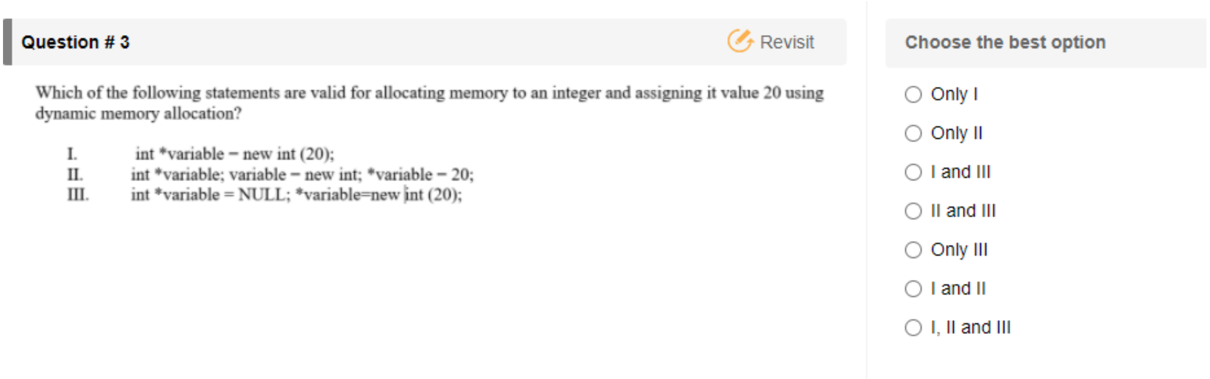
Choose the best option

○ 20, 28

○ 16, 24

○ 8, 16

○ Compiler Error: template parameters cannot have default values



Question # 3

Which of the following statements are valid for allocating memory to an integer and assigning it value 20 using dynamic memory allocation?

```
I.     int *variable = new int (20);
II.    int *variable; variable = new int; *variable = 20;
III.   int *variable = NULL; *variable=new int (20);
```

Choose the best option

○ Only I

○ Only II

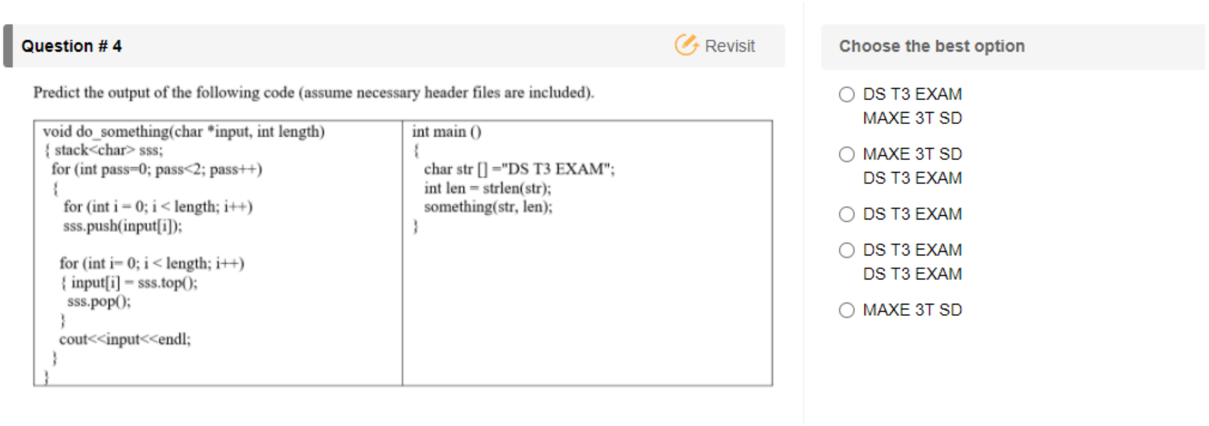
○ I and III

○ II and III

○ Only III

○ I and II

○ I, II and III



Question # 4

Predict the output of the following code (assume necessary header files are included).

```
void do_something(char *input, int length)
{ stack<char> sss;
  for (int pass=0; pass<2; pass++)
  {
    for (int i = 0; i < length; i++)
    sss.push(input[i]);

   for (int i= 0; i < length; i++)
   { input[i] = sss.top();
     sss.pop();
   }
   cout<<input<<endl;
  }
}
```

```
int main ()
{
  char str [] ="DS T3 EXAM";
  int len = strlen(str);
  something(str, len);
}
```

Choose the best option

○ DS T3 EXAM
MAXE 3T SD

○ MAXE 3T SD
DS T3 EXAM

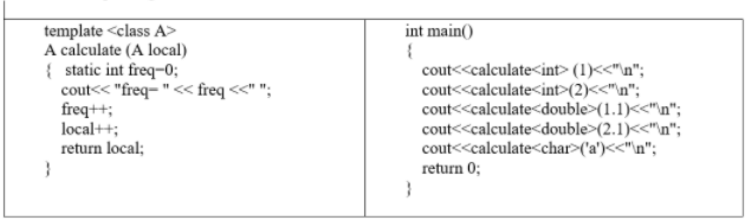
○ DS T3 EXAM

○ DS T3 EXAM
DS T3 EXAM

○ MAXE 3T SD

Question # 2 Revisit

Predict the output of given code:



```
template <class A>
A calculate (A local)
{   static int freq=0;
    cout<< "freq= " << freq <<" ";
    freq++;
    local++;
    return local;
}
```

```
int main()
{
    cout<<calculate<int> (1)<<"\n";
    cout<<calculate<int>(2)<<"\n";
    cout<<calculate<double>(1.1)<<"\n";
    cout<<calculate<double>(2.1)<<"\n";
    cout<<calculate<char>('a')<<"\n";
    return 0;
}
```

Choose the best option

○ freq= 0 2
freq= 1 3
freq= 2 2.1
freq= 3 3.1
freq= 4 b

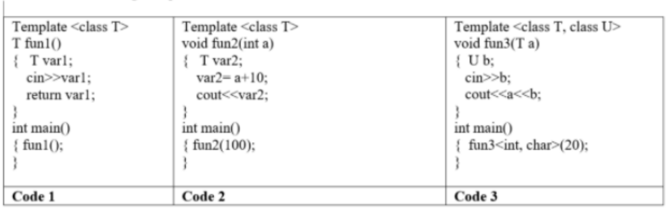
○ freq= 0 2
freq= 0 3
freq= 1 2.1
freq= 1 3.1
freq= 2 b

○ freq= 0 1
freq= 1 2
freq= 2 1.1
freq= 3 1.1
freq= 4 a

○ freq= 0 2
freq= 1 3
freq= 0 2.1
freq= 1 3.1
freq= 0 b

Question # 3 Revisit

Consider the following Template declarations. Which of these declarations are valid?



```
Template <class T>
T fun1()
{   T var1;
    cin>>var1;
    return var1;
}
int main()
{ fun1();
}
```

Code 1

```
Template <class T>
void fun2(int a)
{   T var2;
    var2= a+10;
    cout<<var2;
}
int main()
{ fun2(100);
}
```

Code 2

```
Template <class T, class U>
void fun3(T a)
{  U b;
   cin>>b;
   cout<<a<<b;
}
int main()
{  fun3<int, char>(20);
}
```

Code 3

Choose the best option

○ Code 1 and Code 3 are invalid, Code 2 is valid.

○ Code 2 and Code 3 are valid, Code 1 is invalid.

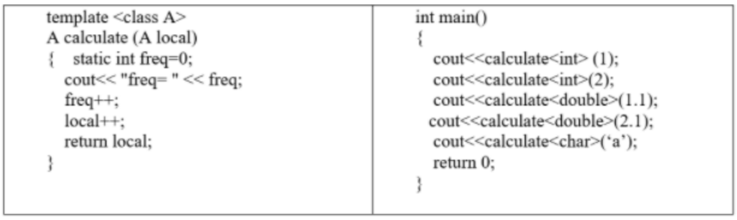
○ Code 1 and code 2 are invalid, Code 3 is valid.

○ Code 1 and Code 2 are valid, Code 3 is invalid.

○ All are invalid.

Question # 4 Revisit

Consider the following function template and the main() function. How many instances of calculate() function template are created in memory?



```
template <class A>
A calculate (A local)
{   static int freq=0;
    cout<< "freq= " << freq;
    freq++;
    local++;
    return local;
}
```

```
int main()
{
    cout<<calculate<int> (1);
    cout<<calculate<int>(2);
    cout<<calculate<double>(1.1);
   cout<<calculate<double>(2.1);
    cout<<calculate<char>(‘a’);
    return 0;
}
```

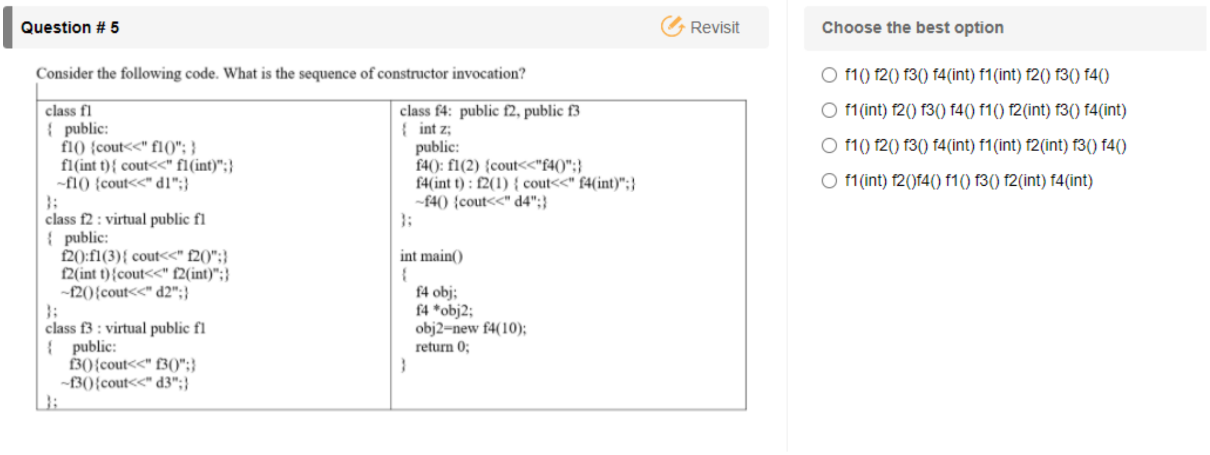
Choose the best option

○ 3

○ 5

○ 1

○ None of the above



Question # 5

Consider the following code. What is the sequence of constructor invocation?

```
class f1
{   public:
    f1() {cout<<" f1()"; }
    f1(int t){ cout<<" f1(int)";}
   ~f1() {cout<<" d1";}
};
class f2 : virtual public f1
{   public:
    f2():f1(3){ cout<<" f2()";}
    f2(int t){cout<<" f2(int)";}
    ~f2(){cout<<" d2";}
};
class f3 : virtual public f1
{     public:
      f3(){cout<<" f3()";}
    ~f3(){cout<<" d3";}
};
```

```
class f4:  public f2, public f3
{   int z;
    public:
    f4(): f1(2) {cout<<"f4()";}
    f4(int t) : f2(1) { cout<<" f4(int)";}
    ~f4() {cout<<" d4";}
};

int main()
{
    f4 obj;
    f4 *obj2;
    obj2=new f4(10);
    return 0;
}
```

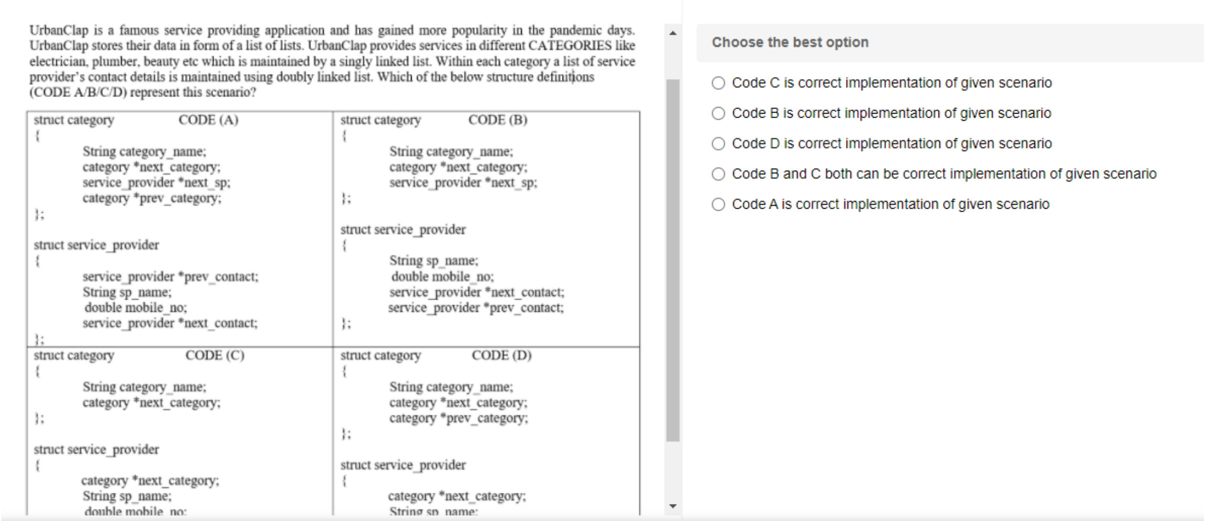
Choose the best option

○ f1() f2() f3() f4(int) f1(int) f2() f3() f4()

○ f1(int) f2() f3() f4() f1() f2(int) f3() f4(int)

○ f1() f2() f3() f4(int) f1(int) f2(int) f3() f4()

○ f1(int) f2()f4() f1() f3() f2(int) f4(int)



UrbanClap is a famous service providing application and has gained more popularity in the pandemic days. UrbanClap stores their data in form of a list of lists. UrbanClap provides services in different CATEGORIES like electrician, plumber, beauty etc which is maintained by a singly linked list. Within each category a list of service provider's contact details is maintained using doubly linked list. Which of the below structure definitions (CODE A/B/C/D) represent this scenario?

```
struct category              CODE (A)
{
            String category_name;
            category *next_category;
            service_provider *next_sp;
            category *prev_category;
};

struct service_provider
{
            service_provider *prev_contact;
            String sp_name;
             double mobile_no;
            service_provider *next_contact;
};
```

```
struct category            CODE (B)
{
            String category_name;
            category *next_category;
            service_provider *next_sp;
};

struct service_provider
{
            String sp_name;
             double mobile_no;
            service_provider *next_contact;
           service_provider *prev_contact;
};
```

```
struct category               CODE (C)
{
            String category_name;
            category *next_category;
};

struct service_provider
{
           category *next_category;
            String sp_name;
             double mobile_no;
```

```
struct category             CODE (D)
{
            String category_name;
            category *next_category;
            category *prev_category;
};

struct service_provider
{
           category *next_category;
           String sp_name;
```

Choose the best option

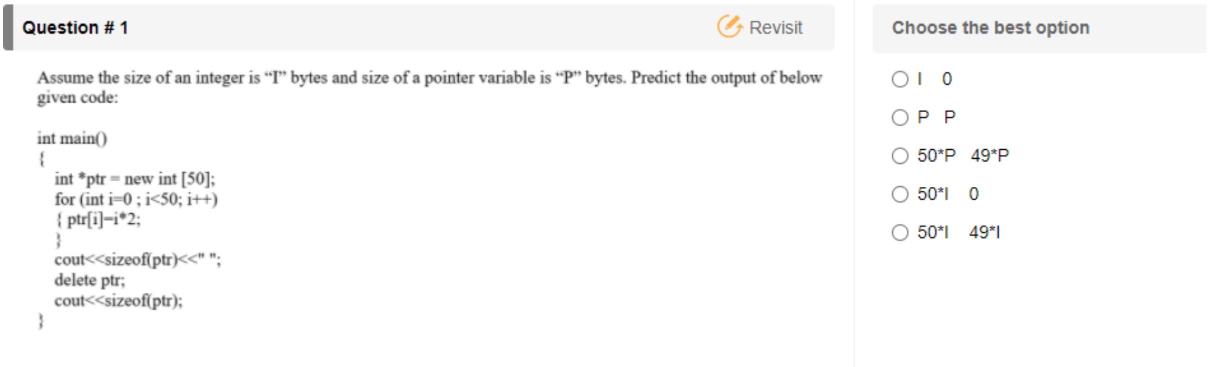
○ Code C is correct implementation of given scenario

○ Code B is correct implementation of given scenario

○ Code D is correct implementation of given scenario

○ Code B and C both can be correct implementation of given scenario

○ Code A is correct implementation of given scenario



Question # 1

Assume the size of an integer is "I" bytes and size of a pointer variable is "P" bytes. Predict the output of below given code:

```
int main()
{
  int *ptr = new int [50];
  for (int i=0 ; i<50; i++)
  { ptr[i]=i*2;
  }
  cout<<sizeof(ptr)<<" ";
  delete ptr;
  cout<<sizeof(ptr);
}
```

Choose the best option

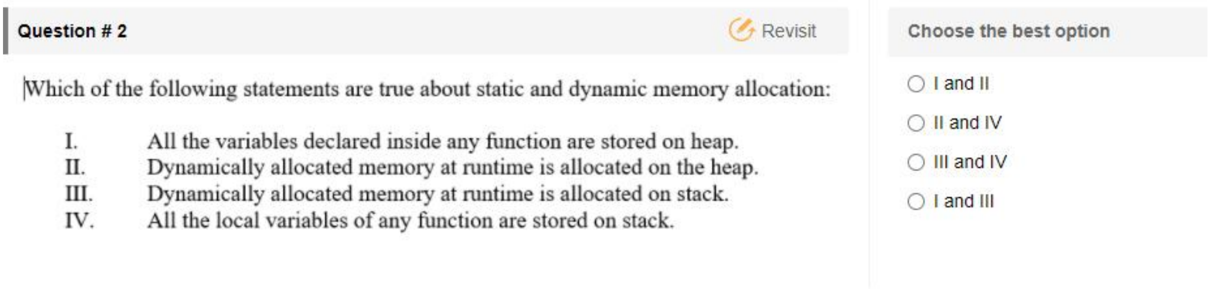
○ I   0

○ P   P

○ 50*P   49*P

○ 50*I   0

○ 50*I   49*I



Question # 2

Which of the following statements are true about static and dynamic memory allocation:

I. All the variables declared inside any function are stored on heap.
II. Dynamically allocated memory at runtime is allocated on the heap.
III. Dynamically allocated memory at runtime is allocated on stack.
IV. All the local variables of any function are stored on stack.

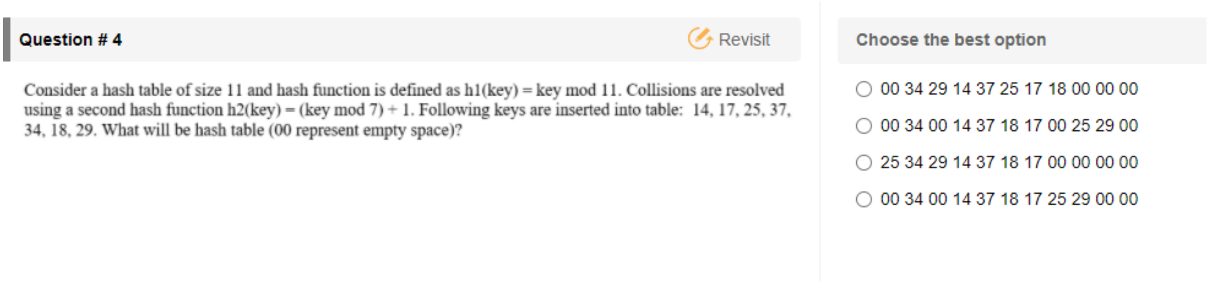
Choose the best option

○ I and II

○ II and IV

○ III and IV

○ I and III



Question # 4

Consider a hash table of size 11 and hash function is defined as h1(key) = key mod 11. Collisions are resolved using a second hash function h2(key) = (key mod 7) + 1. Following keys are inserted into table: 14, 17, 25, 37, 34, 18, 29. What will be hash table (00 represent empty space)?

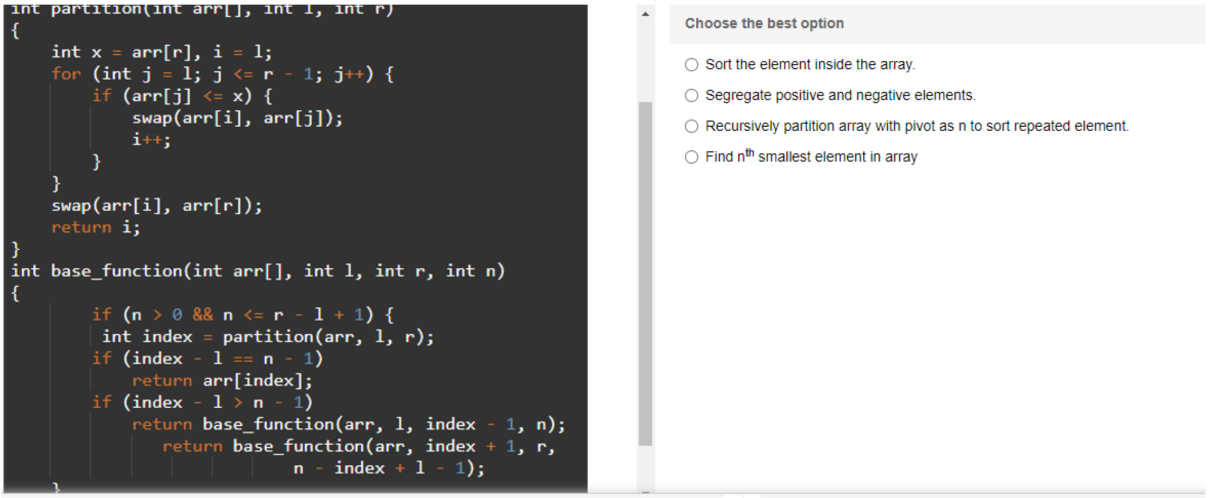
Choose the best option

○ 00 34 29 14 37 25 17 18 00 00 00

○ 00 34 00 14 37 18 17 00 25 29 00

○ 25 34 29 14 37 18 17 00 00 00 00

○ 00 34 00 14 37 18 17 25 29 00 00



```
int partition(int arr[], int l, int r)
{
    int x = arr[r], i = l;
    for (int j = l; j <= r - 1; j++) {
        if (arr[j] <= x) {
            swap(arr[i], arr[j]);
            i++;
        }
    }
    swap(arr[i], arr[r]);
    return i;
}
int base_function(int arr[], int l, int r, int n)
{
        if (n > 0 && n <= r - l + 1) {
         int index = partition(arr, l, r);
        if (index - l == n - 1)
            return arr[index];
        if (index - l > n - 1)
            return base_function(arr, l, index - 1, n);
               return base_function(arr, index + 1, r,
                                    n - index + l - 1);
    }
```

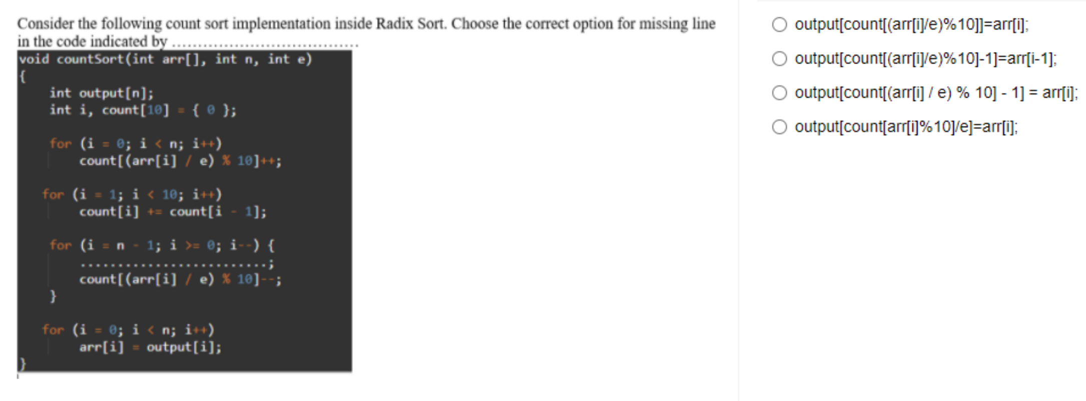
Choose the best option

○ Sort the element inside the array.

○ Segregate positive and negative elements.

○ Recursively partition array with pivot as n to sort repeated element.

○ Find n^th smallest element in array



Consider the following count sort implementation inside Radix Sort. Choose the correct option for missing line in the code indicated by ………………………………

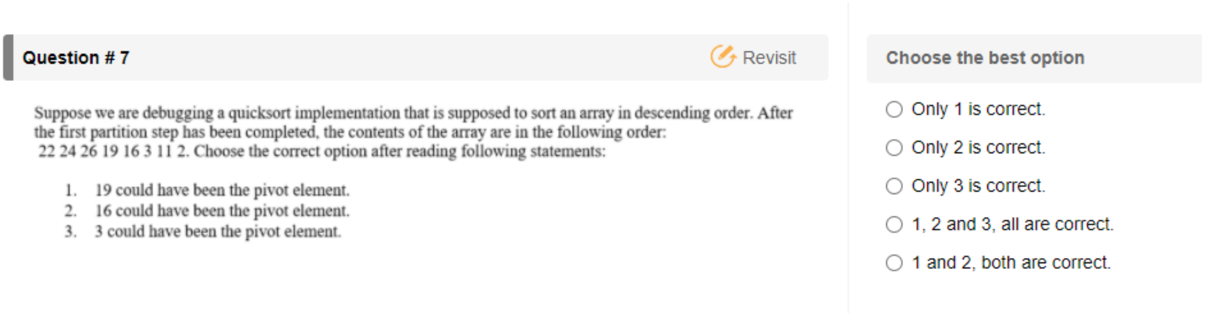
```
void countSort(int arr[], int n, int e)
{
    int output[n];
    int i, count[10] = { 0 };

    for (i = 0; i < n; i++)
        count[(arr[i] / e) % 10]++;

   for (i = 1; i < 10; i++)
        count[i] += count[i - 1];

    for (i = n - 1; i >= 0; i--) {
        .........................;
        count[(arr[i] / e) % 10]--;
    }

   for (i = 0; i < n; i++)
        arr[i] = output[i];
}
```

○ output[count[(arr[i]/e)%10]]=arr[i];

○ output[count[(arr[i]/e)%10]-1]=arr[i-1];

○ output[count[(arr[i] / e) % 10] - 1] = arr[i];

○ output[count[arr[i]%10]/e]=arr[i];



Question # 7

Suppose we are debugging a quicksort implementation that is supposed to sort an array in descending order. After the first partition step has been completed, the contents of the array are in the following order:
22 24 26 19 16 3 11 2. Choose the correct option after reading following statements:

1. 19 could have been the pivot element.
2. 16 could have been the pivot element.
3. 3 could have been the pivot element.

Choose the best option

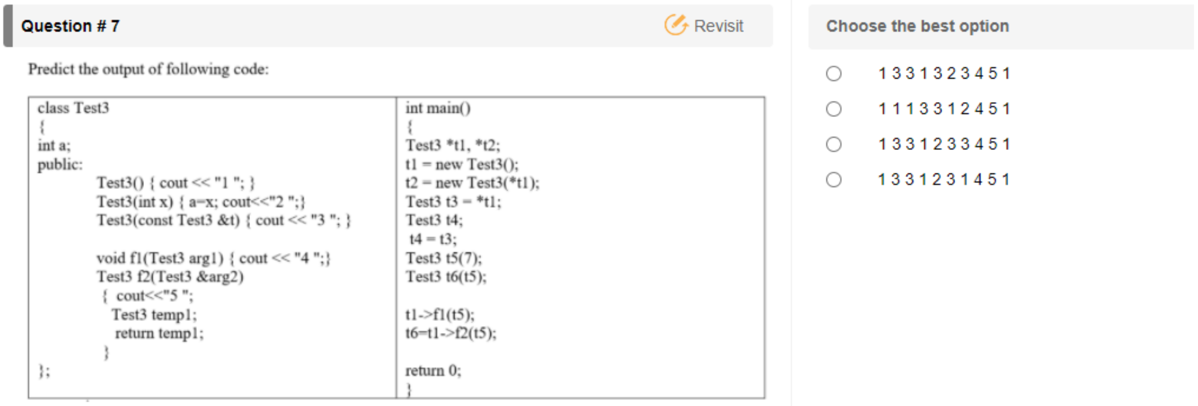
○ Only 1 is correct.

○ Only 2 is correct.

○ Only 3 is correct.

○ 1, 2 and 3, all are correct.

○ 1 and 2, both are correct.



Question # 7

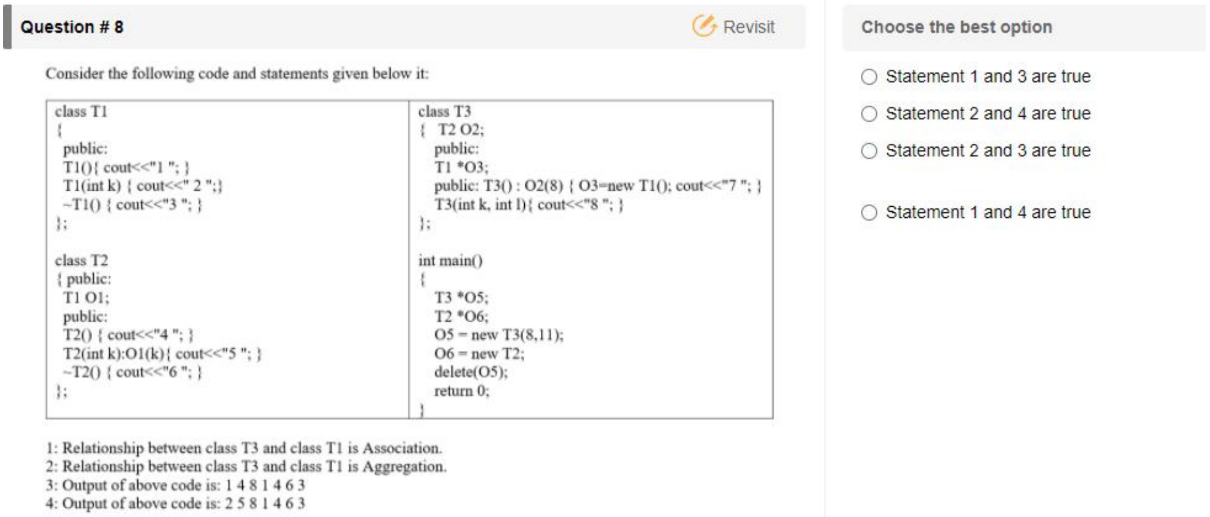
Predict the output of following code:

```
class Test3
{
int a;
public:
          Test3() { cout << "1 "; }
          Test3(int x) { a=x; cout<<"2 ";}
          Test3(const Test3 &t) { cout << "3 "; }

          void f1(Test3 arg1) { cout << "4 ";}
          Test3 f2(Test3 &arg2)
           {  cout<<"5 ";
             Test3 temp1;
              return temp1;
           }
};
```

```
int main()
{
Test3 *t1, *t2;
t1 = new Test3();
t2 = new Test3(*t1);
Test3 t3 = *t1;
Test3 t4;
 t4 = t3;
Test3 t5(7);
Test3 t6(t5);

t1->f1(t5);
t6=t1->f2(t5);

return 0;
}
```

Choose the best option

- ○ 1 3 3 1 3 2 3 4 5 1
- ○ 1 1 1 3 3 1 2 4 5 1
- ○ 1 3 3 1 2 3 3 4 5 1
- ○ 1 3 3 1 2 3 1 4 5 1



Question # 8

Consider the following code and statements given below it:

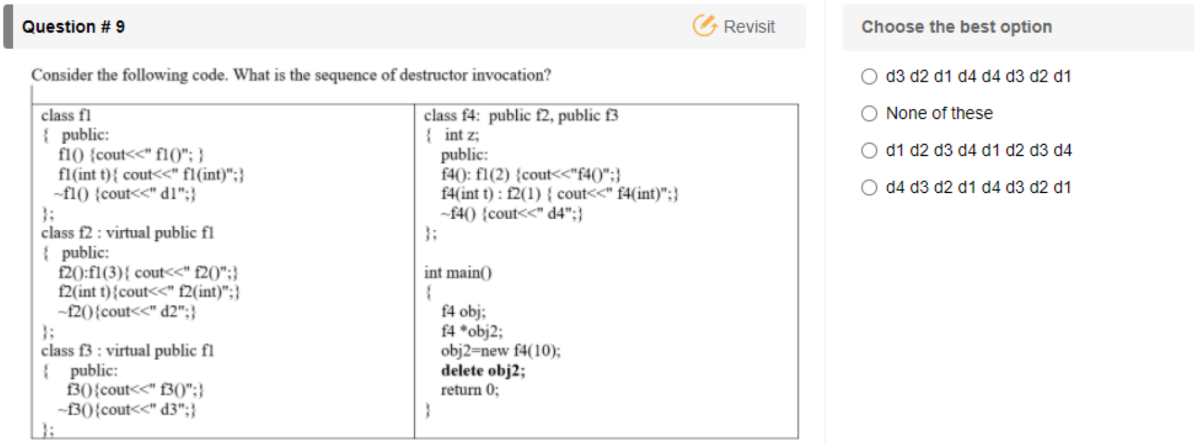
```
class T1
{
 public:
 T1(){ cout<<"1 "; }
 T1(int k) { cout<<" 2 ";}
 ~T1() { cout<<"3 "; }
};

class T2
{ public:
 T1 O1;
 public:
 T2() { cout<<"4 "; }
 T2(int k):O1(k){ cout<<"5 "; }
 ~T2() { cout<<"6 "; }
};
```

```
class T3
{  T2 O2;
   public:
   T1 *O3;
   public: T3() : O2(8) { O3=new T1(); cout<<"7 "; }
   T3(int k, int l){ cout<<"8 "; }
};

int main()
{
   T3 *O5;
   T2 *O6;
   O5 = new T3(8,11);
   O6 = new T2;
   delete(O5);
   return 0;
}
```

1: Relationship between class T3 and class T1 is Association.
2: Relationship between class T3 and class T1 is Aggregation.
3: Output of above code is: 1 4 8 1 4 6 3
4: Output of above code is: 2 5 8 1 4 6 3

Choose the best option

- ○ Statement 1 and 3 are true
- ○ Statement 2 and 4 are true
- ○ Statement 2 and 3 are true
- ○ Statement 1 and 4 are true



Question # 9

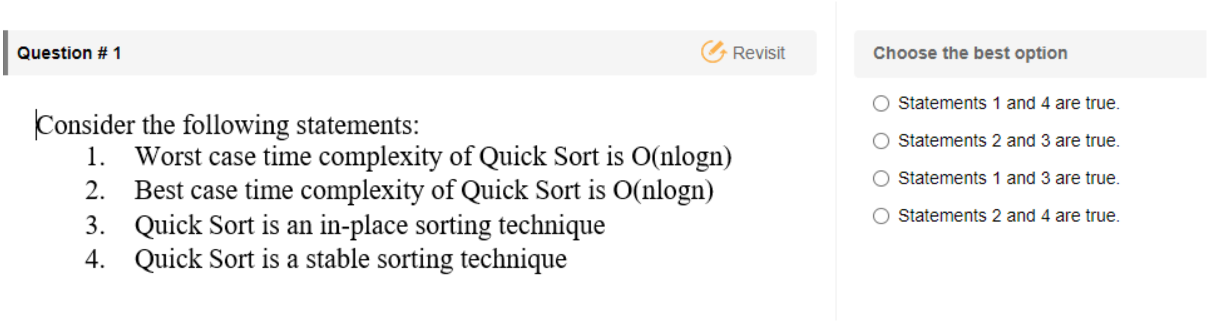
Consider the following code. What is the sequence of destructor invocation?

```
class f1
{   public:
    f1() {cout<<" f1()"; }
    f1(int t){ cout<<" f1(int)";}
   ~f1() {cout<<" d1";}
};
class f2 : virtual public f1
{   public:
    f2():f1(3){ cout<<" f2()";}
    f2(int t){cout<<" f2(int)";}
    ~f2(){cout<<" d2";}
};
class f3 : virtual public f1
{     public:
      f3(){cout<<" f3()";}
    ~f3(){cout<<" d3";}
};
```

```
class f4:  public f2, public f3
{   int z;
    public:
    f4(): f1(2) {cout<<"f4()";}
    f4(int t) : f2(1) { cout<<" f4(int)";}
    ~f4() {cout<<" d4";}
};

int main()
{
    f4 obj;
    f4 *obj2;
    obj2=new f4(10);
    delete obj2;
    return 0;
}
```

Choose the best option

- ○ d3 d2 d1 d4 d4 d3 d2 d1
- ○ None of these
- ○ d1 d2 d3 d4 d1 d2 d3 d4
- ○ d4 d3 d2 d1 d4 d3 d2 d1



Question # 1

Revisit

Consider the following statements:

1. Worst case time complexity of Quick Sort is O(nlogn)
2. Best case time complexity of Quick Sort is O(nlogn)
3. Quick Sort is an in-place sorting technique
4. Quick Sort is a stable sorting technique

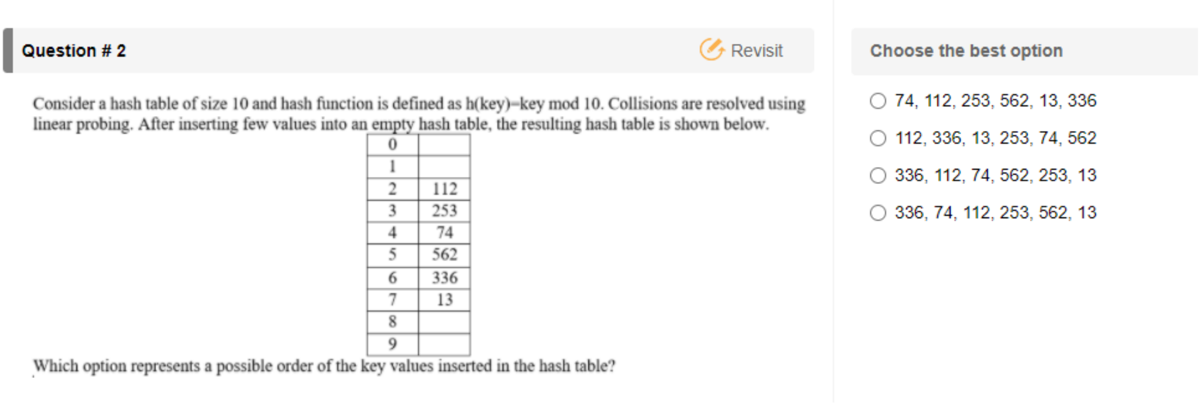
Choose the best option

○ Statements 1 and 4 are true.

○ Statements 2 and 3 are true.

○ Statements 1 and 3 are true.

○ Statements 2 and 4 are true.



Question # 2

Revisit

Consider a hash table of size 10 and hash function is defined as h(key)=key mod 10. Collisions are resolved using linear probing. After inserting few values into an empty hash table, the resulting hash table is shown below.

0	
1	
2	112
3	253
4	74
5	562
6	336
7	13
8	
9	

Which option represents a possible order of the key values inserted in the hash table?

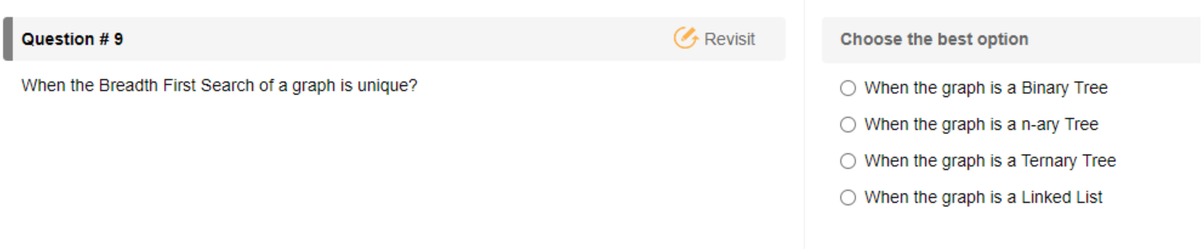
Choose the best option

○ 74, 112, 253, 562, 13, 336

○ 112, 336, 13, 253, 74, 562

○ 336, 112, 74, 562, 253, 13

○ 336, 74, 112, 253, 562, 13



Question # 9

Revisit

When the Breadth First Search of a graph is unique?

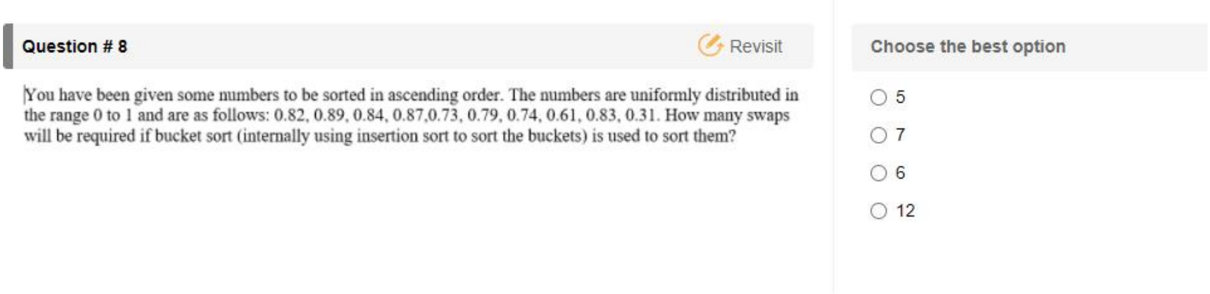
Choose the best option

○ When the graph is a Binary Tree

○ When the graph is a n-ary Tree

○ When the graph is a Ternary Tree

○ When the graph is a Linked List



Question # 8

Revisit

You have been given some numbers to be sorted in ascending order. The numbers are uniformly distributed in the range 0 to 1 and are as follows: 0.82, 0.89, 0.84, 0.87,0.73, 0.79, 0.74, 0.61, 0.83, 0.31. How many swaps will be required if bucket sort (internally using insertion sort to sort the buckets) is used to sort them?

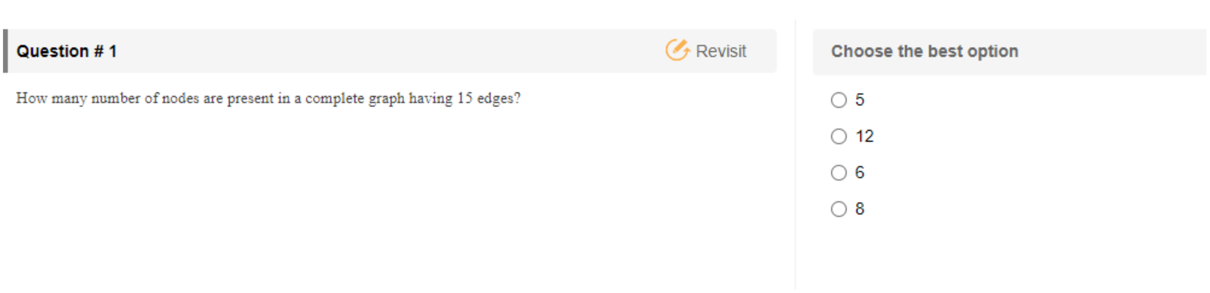
Choose the best option

○ 5

○ 7

○ 6

○ 12



Question # 1

Revisit

How many number of nodes are present in a complete graph having 15 edges?

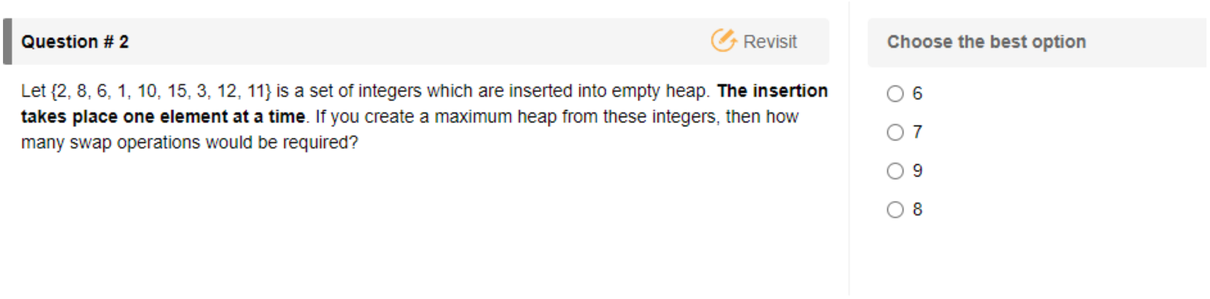
Choose the best option

○ 5

○ 12

○ 6

○ 8



Question # 2

Let {2, 8, 6, 1, 10, 15, 3, 12, 11} is a set of integers which are inserted into empty heap. **The insertion takes place one element at a time**. If you create a maximum heap from these integers, then how many swap operations would be required?

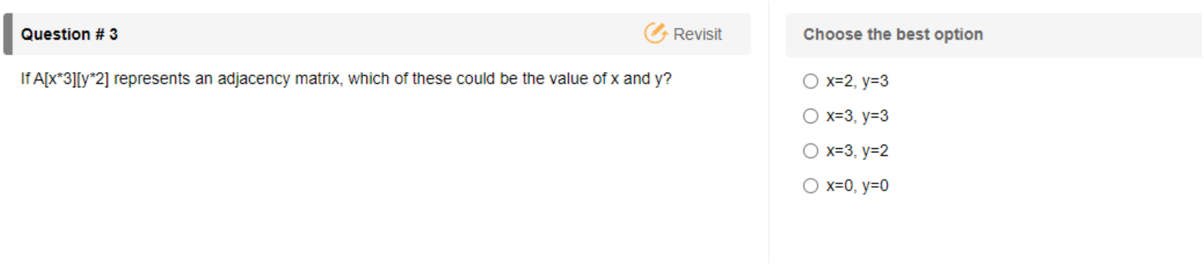
Choose the best option

○ 6

○ 7

○ 9

○ 8



Question # 3

If A[x*3][y*2] represents an adjacency matrix, which of these could be the value of x and y?

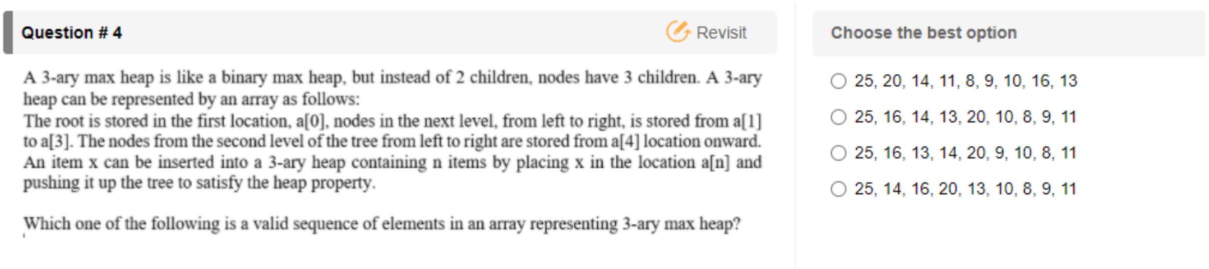
Choose the best option

○ x=2, y=3

○ x=3, y=3

○ x=3, y=2

○ x=0, y=0



Question # 4

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows:
The root is stored in the first location, a[0], nodes in the next level, from left to right, is stored from a[1] to a[3]. The nodes from the second level of the tree from left to right are stored from a[4] location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property.

Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?

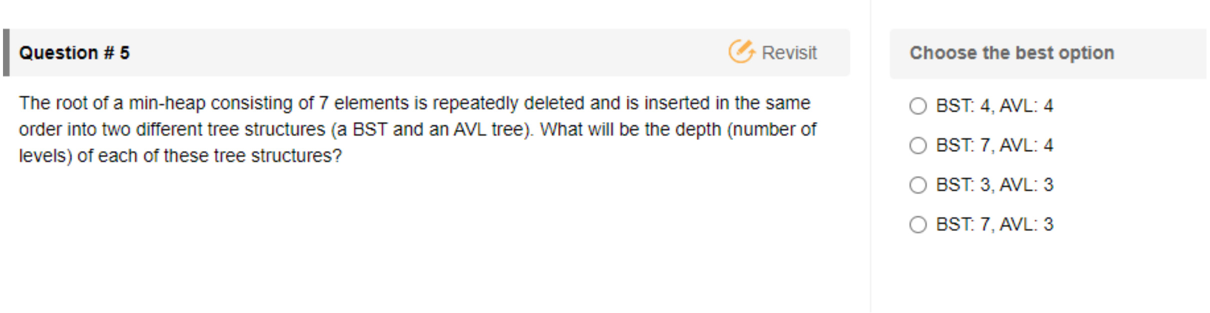
Choose the best option

○ 25, 20, 14, 11, 8, 9, 10, 16, 13

○ 25, 16, 14, 13, 20, 10, 8, 9, 11

○ 25, 16, 13, 14, 20, 9, 10, 8, 11

○ 25, 14, 16, 20, 13, 10, 8, 9, 11



Question # 5

The root of a min-heap consisting of 7 elements is repeatedly deleted and is inserted in the same order into two different tree structures (a BST and an AVL tree). What will be the depth (number of levels) of each of these tree structures?

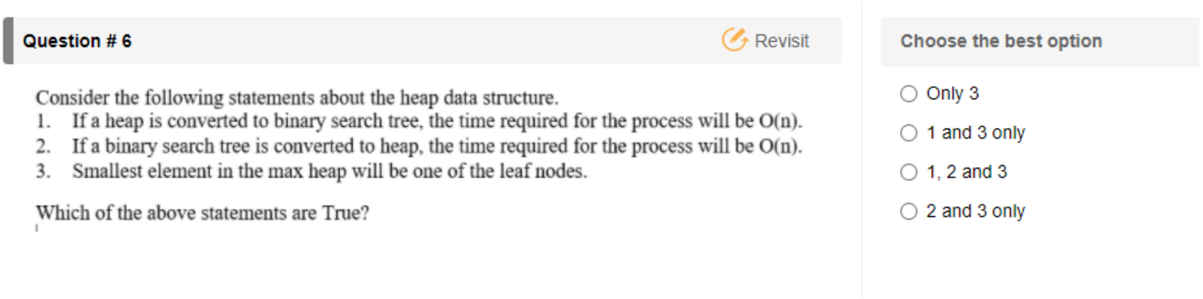
Choose the best option

○ BST: 4, AVL: 4

○ BST: 7, AVL: 4

○ BST: 3, AVL: 3

○ BST: 7, AVL: 3



Question # 6

Consider the following statements about the heap data structure.

1. If a heap is converted to binary search tree, the time required for the process will be O(n).
2. If a binary search tree is converted to heap, the time required for the process will be O(n).
3. Smallest element in the max heap will be one of the leaf nodes.

Which of the above statements are True?

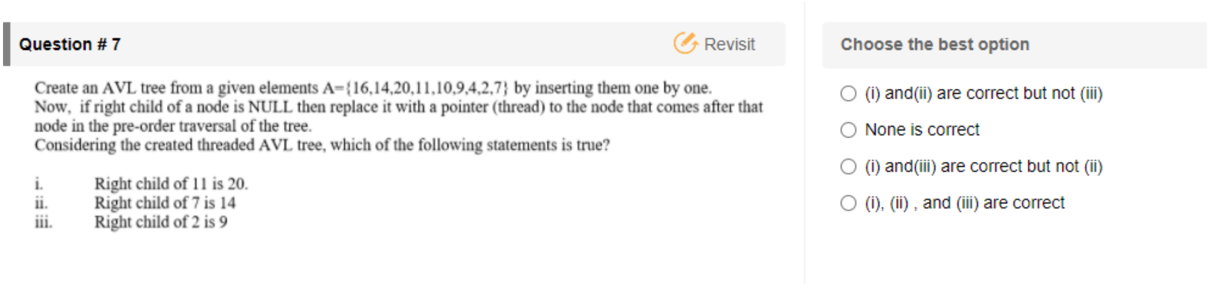
Choose the best option

○ Only 3

○ 1 and 3 only

○ 1, 2 and 3

○ 2 and 3 only



Question # 7

Create an AVL tree from a given elements A={16,14,20,11,10,9,4,2,7} by inserting them one by one.
Now, if right child of a node is NULL then replace it with a pointer (thread) to the node that comes after that node in the pre-order traversal of the tree.
Considering the created threaded AVL tree, which of the following statements is true?

i. Right child of 11 is 20.
ii. Right child of 7 is 14
iii. Right child of 2 is 9

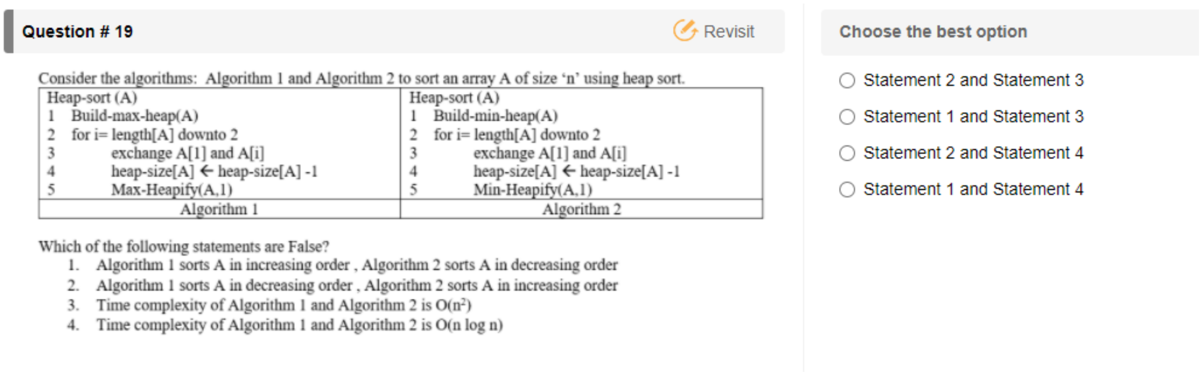
Choose the best option

○ (i) and(ii) are correct but not (iii)

○ None is correct

○ (i) and(iii) are correct but not (ii)

○ (i), (ii) , and (iii) are correct



Question # 19

Consider the algorithms: Algorithm 1 and Algorithm 2 to sort an array A of size 'n' using heap sort.

```
Heap-sort (A)
   Build-max-heap(A)
   for i= length[A] downto 2
         exchange A[1] and A[i]
         heap-size[A] ← heap-size[A] -1
         Max-Heapify(A,1)
```

Algorithm 1

```
Heap-sort (A)
   Build-min-heap(A)
   for i= length[A] downto 2
         exchange A[1] and A[i]
         heap-size[A] ← heap-size[A] -1
         Min-Heapify(A,1)
```

Algorithm 2

Which of the following statements are False?

1. Algorithm 1 sorts A in increasing order , Algorithm 2 sorts A in decreasing order
2. Algorithm 1 sorts A in decreasing order , Algorithm 2 sorts A in increasing order
3. Time complexity of Algorithm 1 and Algorithm 2 is $O(n^2)$
4. Time complexity of Algorithm 1 and Algorithm 2 is O(n log n)

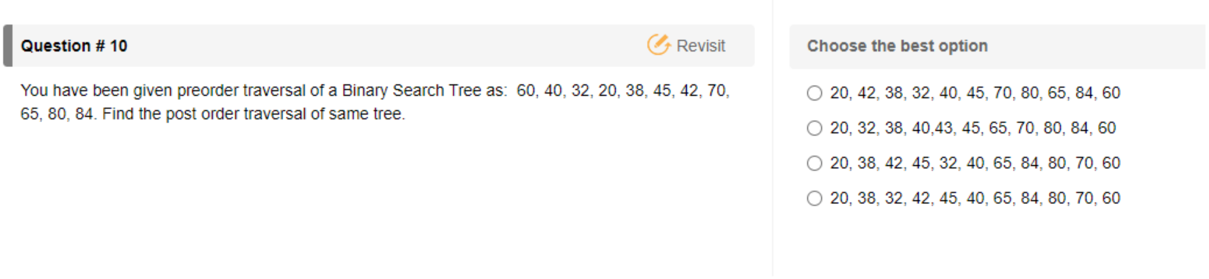
Choose the best option

○ Statement 2 and Statement 3

○ Statement 1 and Statement 3

○ Statement 2 and Statement 4

○ Statement 1 and Statement 4



Question # 10

You have been given preorder traversal of a Binary Search Tree as: 60, 40, 32, 20, 38, 45, 42, 70, 65, 80, 84. Find the post order traversal of same tree.

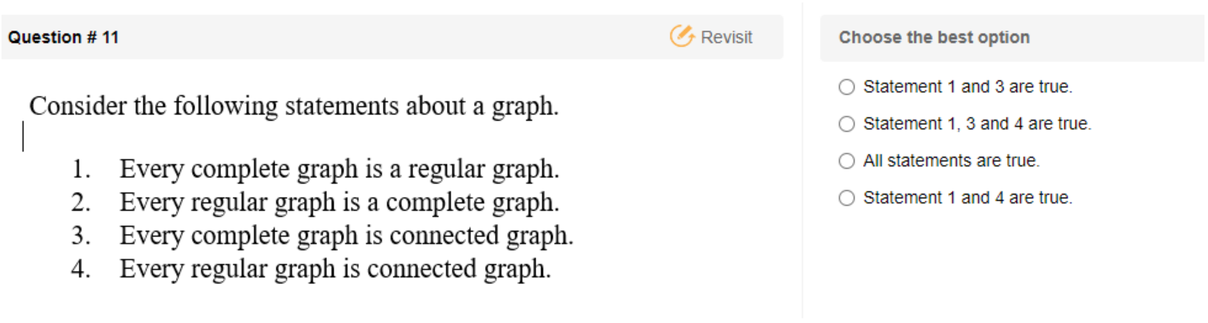
Choose the best option

○ 20, 42, 38, 32, 40, 45, 70, 80, 65, 84, 60

○ 20, 32, 38, 40,43, 45, 65, 70, 80, 84, 60

○ 20, 38, 42, 45, 32, 40, 65, 84, 80, 70, 60

○ 20, 38, 32, 42, 45, 40, 65, 84, 80, 70, 60



Question # 11

Consider the following statements about a graph.

1. Every complete graph is a regular graph.
2. Every regular graph is a complete graph.
3. Every complete graph is connected graph.
4. Every regular graph is connected graph.

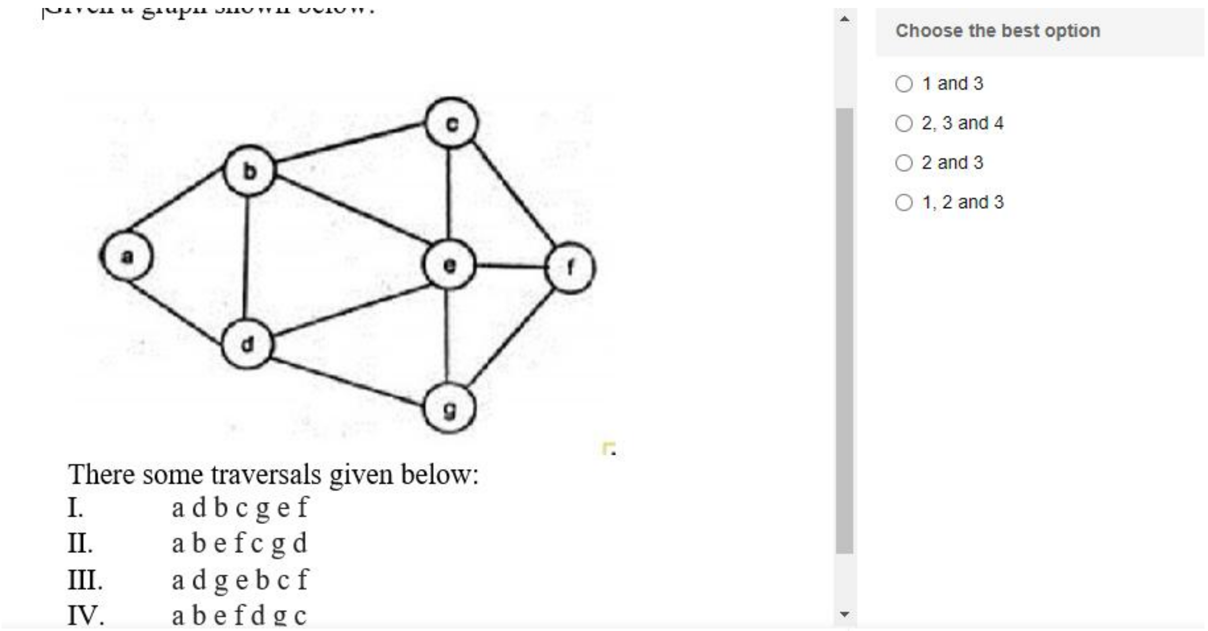
Choose the best option

○ Statement 1 and 3 are true.

○ Statement 1, 3 and 4 are true.

○ All statements are true.

○ Statement 1 and 4 are true.



Given a graph shown below:

There some traversals given below:

I. a d b c g e f

II. a b e f c g d

III. a d g e b c f

IV. a b e f d g c

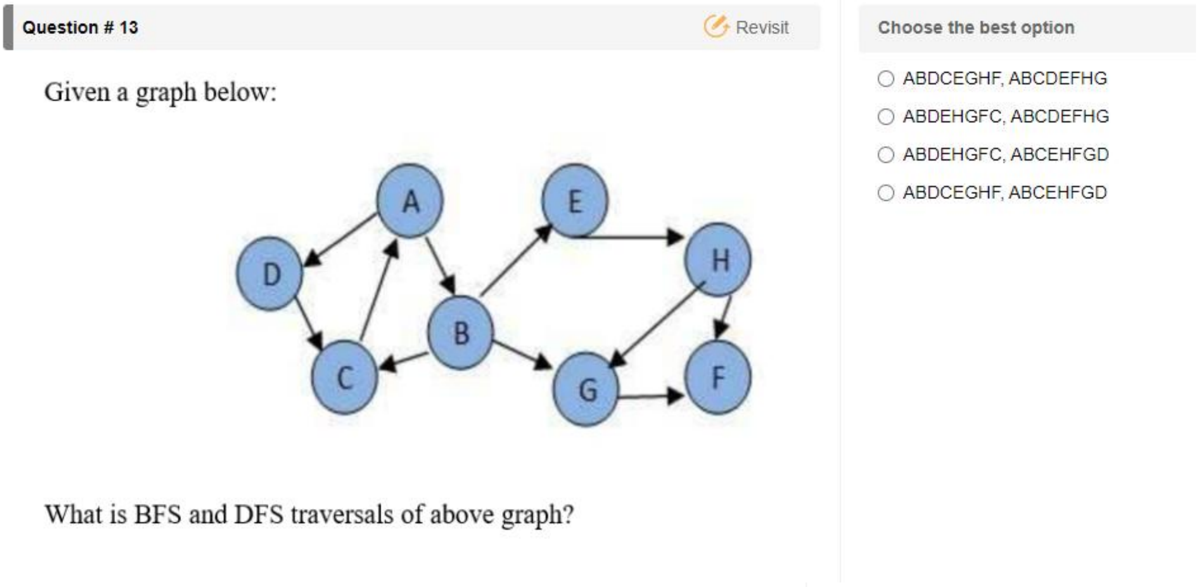
Choose the best option

○ 1 and 3

○ 2, 3 and 4

○ 2 and 3

○ 1, 2 and 3



Question # 13 Revisit

Given a graph below:

What is BFS and DFS traversals of above graph?

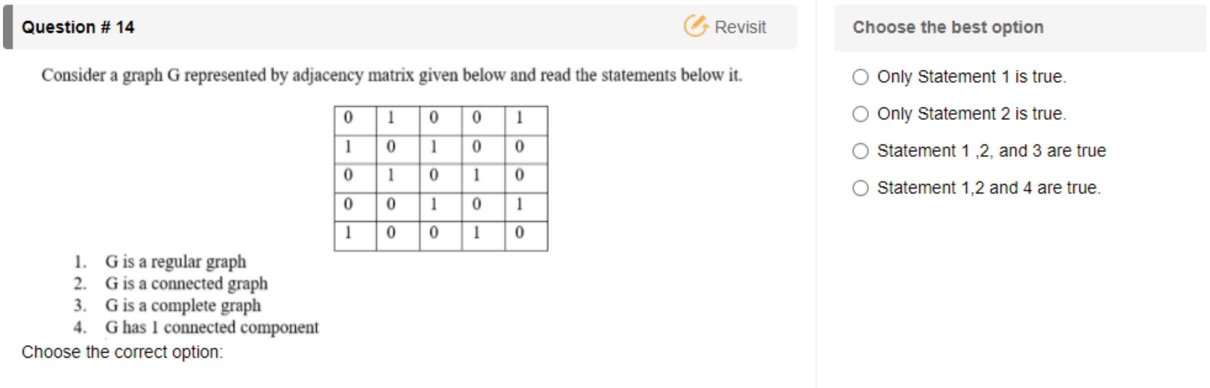
Choose the best option

○ ABDCEGHF, ABCDEFHG

○ ABDEHGFC, ABCDEFHG

○ ABDEHGFC, ABCEHFGD

○ ABDCEGHF, ABCEHFGD



Question # 14 Revisit

Consider a graph G represented by adjacency matrix given below and read the statements below it.

0	1	0	0	1
1	0	1	0	0
0	1	0	1	0
0	0	1	0	1
1	0	0	1	0

1. G is a regular graph
2. G is a connected graph
3. G is a complete graph
4. G has 1 connected component

Choose the correct option:

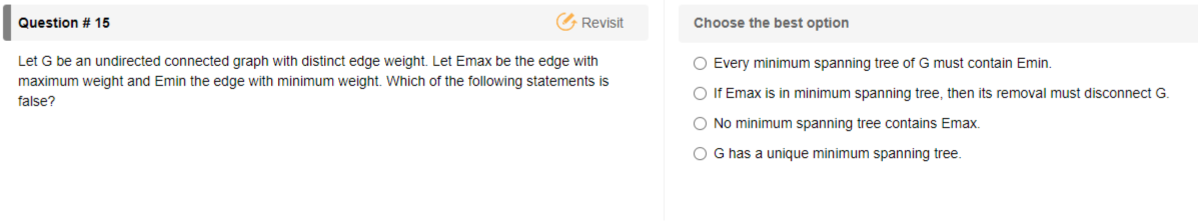
Choose the best option

○ Only Statement 1 is true.

○ Only Statement 2 is true.

○ Statement 1 ,2, and 3 are true

○ Statement 1,2 and 4 are true.



Question # 15 Revisit

Let G be an undirected connected graph with distinct edge weight. Let Emax be the edge with maximum weight and Emin the edge with minimum weight. Which of the following statements is false?

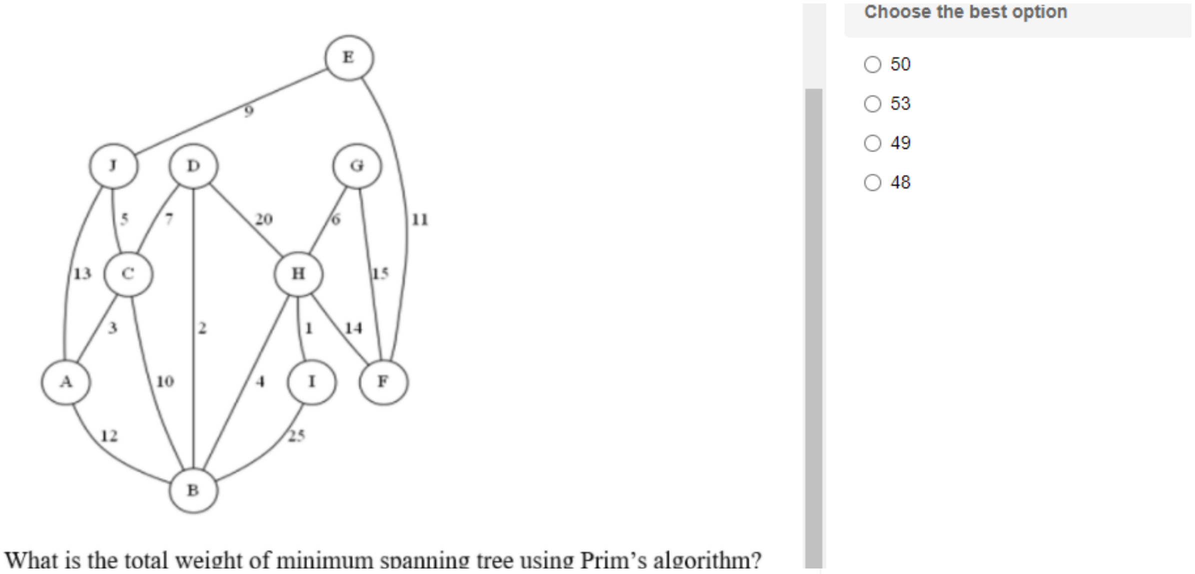
Choose the best option

○ Every minimum spanning tree of G must contain Emin.

○ If Emax is in minimum spanning tree, then its removal must disconnect G.

○ No minimum spanning tree contains Emax.

○ G has a unique minimum spanning tree.



What is the total weight of minimum spanning tree using Prim's algorithm?

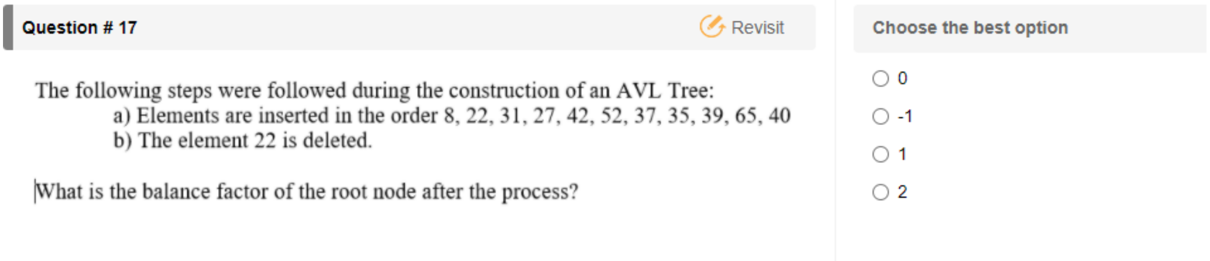
Choose the best option

○ 50

○ 53

○ 49

○ 48



Question # 17 Revisit

The following steps were followed during the construction of an AVL Tree:

a) Elements are inserted in the order 8, 22, 31, 27, 42, 52, 37, 35, 39, 65, 40

b) The element 22 is deleted.

What is the balance factor of the root node after the process?

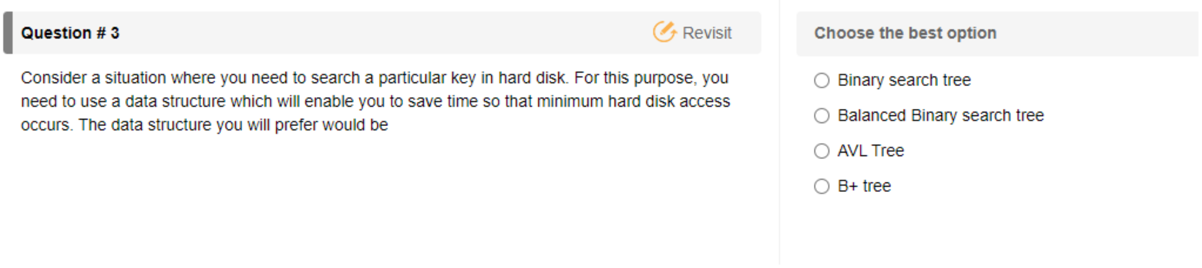
Choose the best option

○ 0

○ -1

○ 1

○ 2



Question # 3 Revisit

Consider a situation where you need to search a particular key in hard disk. For this purpose, you need to use a data structure which will enable you to save time so that minimum hard disk access occurs. The data structure you will prefer would be

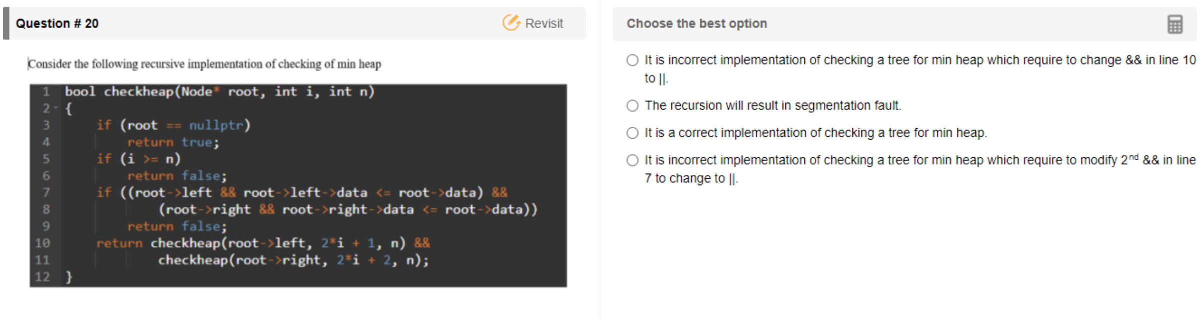
Choose the best option

○ Binary search tree

○ Balanced Binary search tree

○ AVL Tree

○ B+ tree



Question # 20 Revisit

Consider the following recursive implementation of checking of min heap

```
1   bool checkheap(Node* root, int i, int n)
2   {
3       if (root == nullptr)
4           return true;
5       if (i >= n)
6           return false;
7       if ((root->left && root->left->data <= root->data) &&
8               (root->right && root->right->data <= root->data))
9           return false;
10      return checkheap(root->left, 2*i + 1, n) &&
11              checkheap(root->right, 2*i + 2, n);
12  }
```

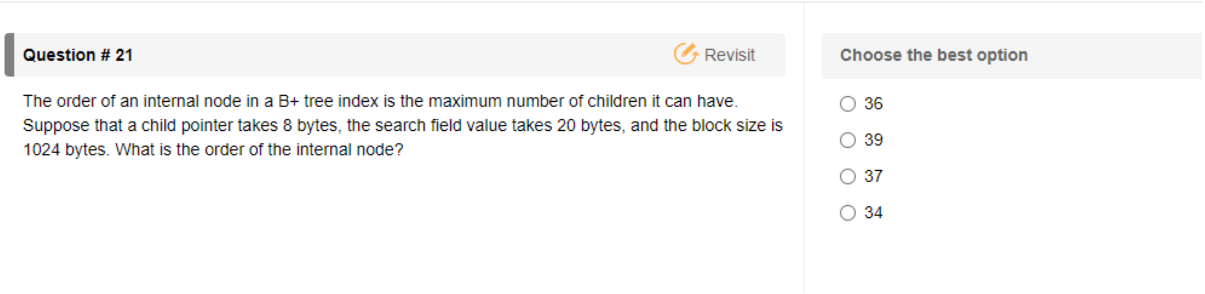
Choose the best option

○ It is incorrect implementation of checking a tree for min heap which require to change && in line 10 to ||.

○ The recursion will result in segmentation fault.

○ It is a correct implementation of checking a tree for min heap.

○ It is incorrect implementation of checking a tree for min heap which require to modify 2nd && in line 7 to change to ||.



Question # 21

The order of an internal node in a B+ tree index is the maximum number of children it can have. Suppose that a child pointer takes 8 bytes, the search field value takes 20 bytes, and the block size is 1024 bytes. What is the order of the internal node?

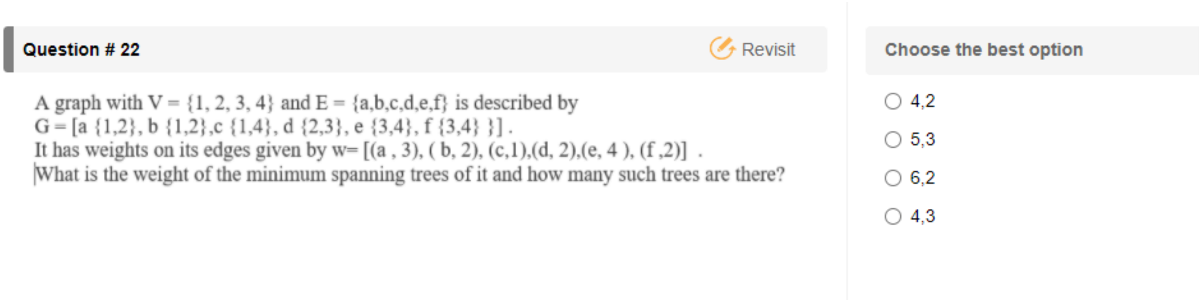
Choose the best option

○ 36

○ 39

○ 37

○ 34



Question # 22

A graph with V = {1, 2, 3, 4} and E = {a,b,c,d,e,f} is described by
G = [a {1,2}, b {1,2},c {1,4}, d {2,3}, e {3,4}, f {3,4} }] .
It has weights on its edges given by w= [(a , 3), (b, 2), (c,1),(d, 2),(e, 4), (f ,2)] .
What is the weight of the minimum spanning trees of it and how many such trees are there?

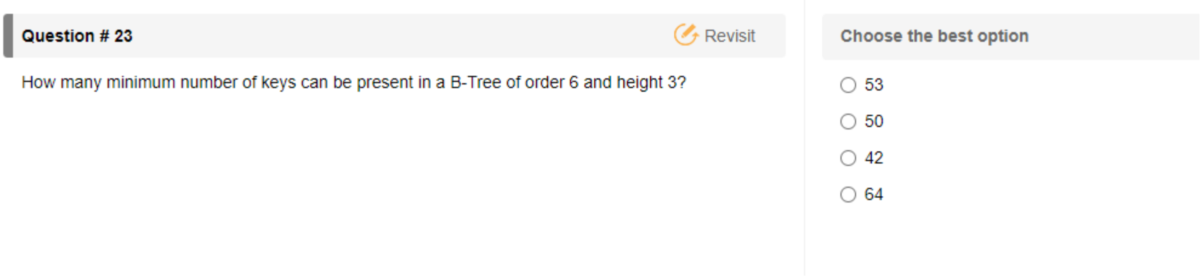
Choose the best option

○ 4,2

○ 5,3

○ 6,2

○ 4,3



Question # 23

How many minimum number of keys can be present in a B-Tree of order 6 and height 3?

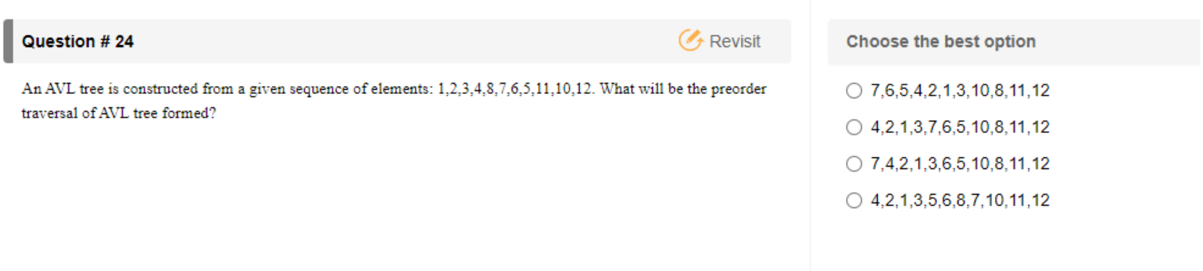
Choose the best option

○ 53

○ 50

○ 42

○ 64



Question # 24

An AVL tree is constructed from a given sequence of elements: 1,2,3,4,8,7,6,5,11,10,12. What will be the preorder traversal of AVL tree formed?

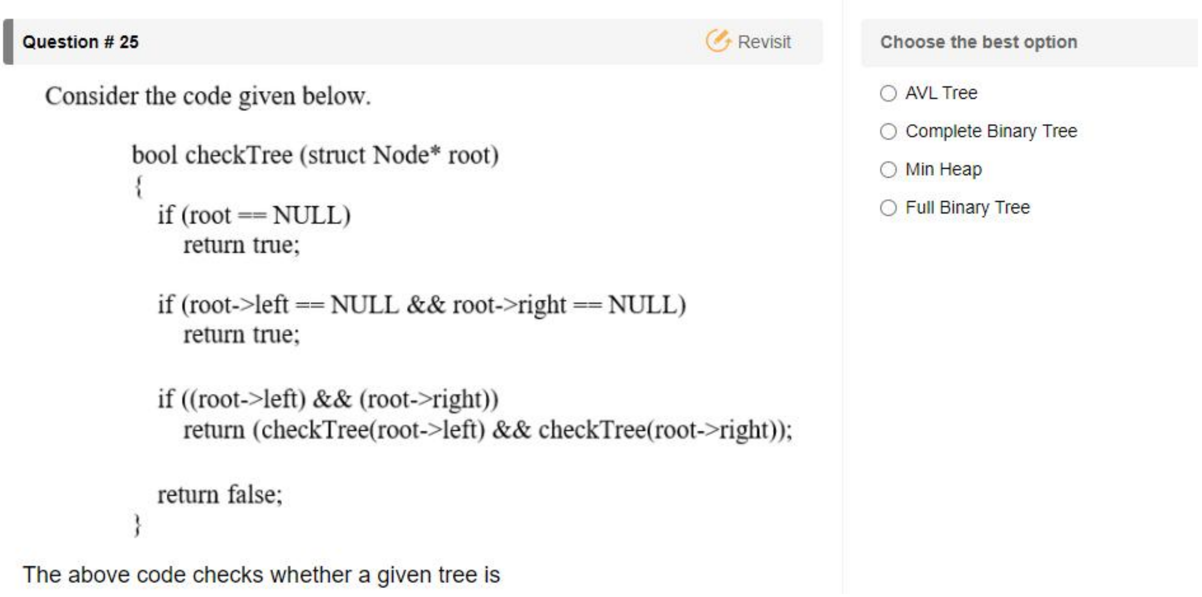
Choose the best option

○ 7,6,5,4,2,1,3,10,8,11,12

○ 4,2,1,3,7,6,5,10,8,11,12

○ 7,4,2,1,3,6,5,10,8,11,12

○ 4,2,1,3,5,6,8,7,10,11,12



Question # 25

Consider the code given below.

```
bool checkTree (struct Node* root)
{
   if (root == NULL)
      return true;

   if (root->left == NULL && root->right == NULL)
      return true;

   if ((root->left) && (root->right))
      return (checkTree(root->left) && checkTree(root->right));

   return false;
}
```

The above code checks whether a given tree is

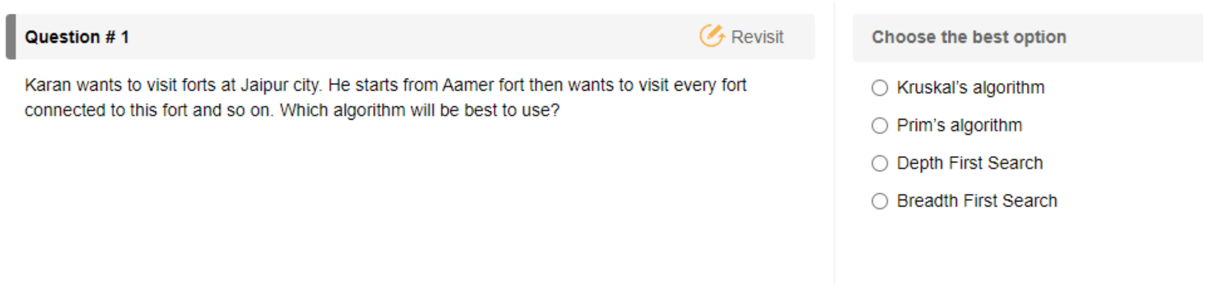
Choose the best option

○ AVL Tree

○ Complete Binary Tree

○ Min Heap

○ Full Binary Tree



Question # 1

Karan wants to visit forts at Jaipur city. He starts from Aamer fort then wants to visit every fort connected to this fort and so on. Which algorithm will be best to use?

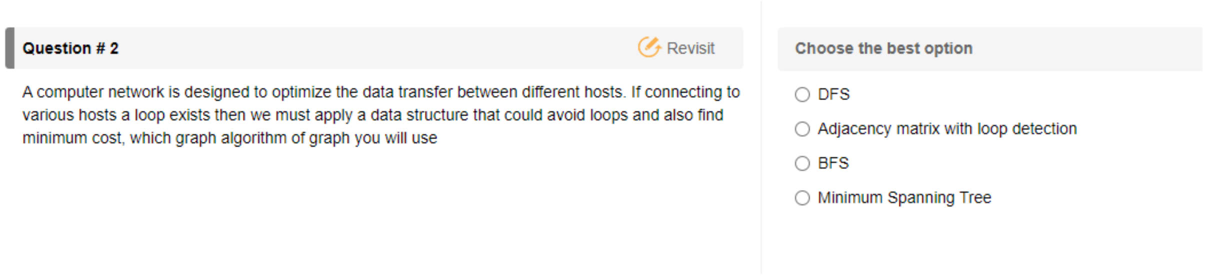
Choose the best option

○ Kruskal's algorithm

○ Prim's algorithm

○ Depth First Search

○ Breadth First Search



Question # 2

A computer network is designed to optimize the data transfer between different hosts. If connecting to various hosts a loop exists then we must apply a data structure that could avoid loops and also find minimum cost, which graph algorithm of graph you will use

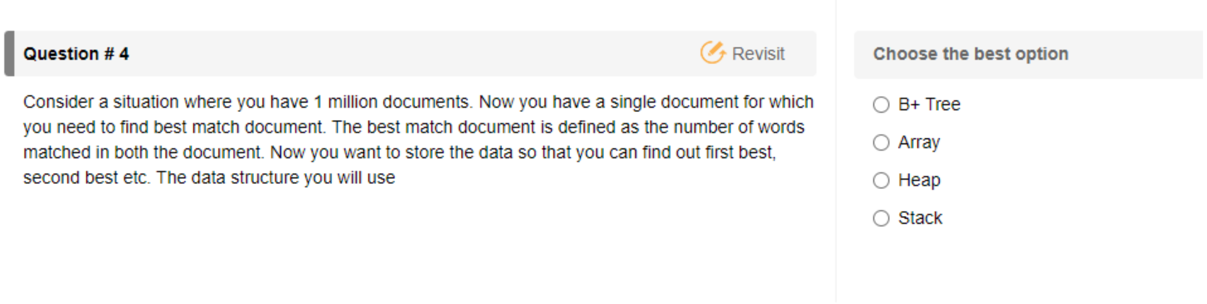
Choose the best option

○ DFS

○ Adjacency matrix with loop detection

○ BFS

○ Minimum Spanning Tree



Question # 4

Consider a situation where you have 1 million documents. Now you have a single document for which you need to find best match document. The best match document is defined as the number of words matched in both the document. Now you want to store the data so that you can find out first best, second best etc. The data structure you will use

Choose the best option

○ B+ Tree

○ Array

○ Heap

○ Stack