**MCQ SECTION  
Q1. What is the output of code? (Score- 2)**

#include <iostream>

#include <deque>

#include <vector>

using namespace std;

int main ()

{

int a[] = {1,5,8,9,3};

deque<int>dq(a, a+5);

dq.push\_back(10);

dq.push\_front(20);

deque<int>::iterator i;

i=dq.begin()+2;

dq.insert(i,15);

int a1[]={7,7,7,7};

dq.insert(dq.begin() , a1 , a1+4 );

for(i=dq.begin();i!=dq.end();i++)

cout<<\*i<<"";

}

* 1. **7 7 7 7 20 1 15 5 8 9 3 10**
  2. 20 1 15 5 8 9 3 10 7 7 7 7
  3. 7 7 7 7 1 15 5 8 9 3 10 20
  4. 7 7 20 1 15 5 8 9 3 10 7 7

**Q2.What is the output of this program?(score-2)**

#include<iostream>

using namespace std;

int main()

{

cout.width(10);

cout.fill('\*');

cout <<"DONE";

}

* 1. **\*\*\*\*\*\*DONE**
  2. DONE
  3. DONE\*\*\*\*\*\*
  4. \*DONE

#### Q3. ios::trunc is used for ? (Score-2)

1. If the file is opened for output operations and it already existed, no action is taken.
2. **If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.**
3. If the file is opened for output operations and it already existed, then a new copy of file is created.
4. None of above

**Q4. Which of the following statement is incorrect? (Score -2)**

a) eof() can be used to check whether we have reached the end of file or not

b) fopen can be used to open the in binary mode

c) **we can only access the file in sequential manner**

d) input stream is used for reading data from the file on the disk.

**Q5. Functions called from within a try block may also throw exception. (Score -2)**

1. False
2. **True**
3. May be
4. None of the above

**CODING SECTION**

*PROBLEM STATEMENT-1(10 marks)*

Q5. You are given a task to create a vector with some elements. Firstly display the elements of vector. Now help a user to display elements of a vector in reverse order. Also display maximum and minimum element in the vector.

**Sample Input Test Case 1:**

4 // (N) size of vector

4 3 6 5 //elements of vector

**Sample Output Test Case 1:**

5 6 3 4 // vector after reverse

6 // maximum element

3 // minimum element

**Sample Input Test Case 2:**

6 //(N) size of vector

0 1 2 3 4 5 // elements of vector

**Sample Output Test Case 2:**

5 4 3 2 1 0 //vector after reverse

5 // maximum element

0 // minimum element

**Constraint**: each input(n) varies as 0<=n<=100 & 0<=N<=100

**Explanation:**

**Sample Input:**

First line denotes size of vector

Second line denote elements to be entered in vector

**Sample Output:**

First line denotes elements of vector after reverse

Second line denotes maximum element

Third line denotes minimum element

**Head:**

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

int size,value;

cin>>size;

vector<int> v;

vector<int>:: iterator i;

**Tail:**

cout<<\*max<<endl<<\*min; //Maximum and Minimum element

return 0;

}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testcase0**  **(sample)**  **(Score-0)**  **Input**  4  4 3 5 6  **Output**  6 5 3 4  6  3 | **Testcase1**  **(sample)**  **(Score-0)**  **Input**  6  0 1 2 3 4 5  **Output**  5 4 3 2 1 0  5  0 | **Testcase2**  **(Score-1)**  **Input**  3  6 7 8  **Output**  8 7 6  8  6 | **Testcase3**  **(Score-1)**  **Input**  5  3 6 8 9 2  **Output**  2 9 8 6 3  9  2 | **Testcase4**  **(Score-1)**  **Input**  6  4 3 2 5 7 6  **Output**  6 7 5 2 3 4  7  2 | **Testcase5**  **(Score-1)**  **Input**  2  6 7  **Output**  7 6  7  6 | **Testcase6**  **(Score-1)**  **Input**  7  1 2 3 4 5 6 7  **Output**  7 6 5 4 3 2 1  7  1 |