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

#include <iostream>

#include <vector>

#include<iterator>

using namespace std;

int main ()

{

vector<int>myvector;

for (int i = 1; i <= 10; i++)

myvector.push\_back(i);

myvector.erase (myvector.begin() + 6);

myvector.erase (myvector.begin(), myvector.begin() + 4);

for (unsigned i = 0; i <myvector.size(); ++i)

cout<<''<<myvector[i];

return 0;

}

1. 5 6 7 8 9
2. **5 6 8 9 10**
3. 6 7 8 9 10
4. 4 5 6 8 9 10

**Q2.What happens in C++ when an exception is thrown and not caught anywhere like following program. (Score- 2)**

#include <iostream>

using namespace std;

int scorn() throw (int)

{

    throw 108;

}

 int main()

{

  scorn();

  return 0;

}

1. **Program terminates abnormally.**
2. Program terminates normally but doesn’t print anything.
3. Compiler Error
4. None of the above

**Q3. Predict the output of the given code.(score-2)**

#include<fstream.h>

#include<iostream.h>

void main()

{

char buffer[100];

ifstream inFile(“comments.txt",ios::nocreate);

while (!inFile.getline(buffer,100)

cout<<buffer;

inFile.close();

}

1. **Compile time error**
2. display content of file
3. write new content to file
4. run time error

**Q4. What do vectors represent? (Score- 2)**

a) Static arrays

b) **Dynamic arrays**

c) Stack

d) Queue

Answer B

Q5. Output of the following code:  
void south\_lebanon(int silk) throw (float,int)  
{ if(silk==2)  
throw silk;  
else if(silk==4)  
throw (char)silk;  
else  
throw (float)silk; }  
int main()  
{ try  
{ south\_lebanon (4);  
south\_lebanon (2); }  
catch(int)  
{ cout<<"Caught integer exception"; }  
catch(char)  
{ cout<<"Caught char exception"; }  
catch(float)  
{ cout<<"Caught float exception";}  
}

1. Caught char exception

Caught integer exception

1. Caught float exception

Caught integer exception

1. Caught char exception
2. **Aborted**

**CODING SECTION**

*PROBLEM STATEMENT-1(10 marks)*

A LIFO structure is to be maintained for a set of elements. A LIFO structure is that in which the element inserted at last will be accessed or removed first. You need to insert and display the elements stored in this structure.

**Sample Input Test Case 1:**

4 //(N) size

2 3 5 6 //Numbers to be enter

**Sample Output Test Case 1:**

6 5 3 2 //displayed elements in LIFO structure

**Sample Input Test Case 2:**

7 //(N) size

1 2 3 4 5 6 7 //Numbers to be eneter

**Sample Output Test Case 2:**

7 6 5 4 3 2 1 //displayed elements in LIFO structure

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

**Explanation:**

**Sample Input:**

First line denotes size of LIFO sturcture

Second line denote element to be entered in it

**Sample Output:**

First line denotes elements of LIFO structure

**Head:**

#include <iostream>

#include <algorithm>

#include <stack>

using namespace std;

int main()

{

stack<int> s;

**Tail:**

s.pop();

}

return 0;

}

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