**MCQ SECTION  
Q1. Output of the following code: (Score- 2)**

#include<iostream>

using namespace std;

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";}

}

a) Caught char exception

Caught integer exception

b) **Aborted**

c) Caught float exception

Caught integer exception

d) Caught char exception

**Q2. Which function return the current position of the get and put pointer in bytes. (Score – 2)**

a) tellg( )

b) tellp( )

c) **Both A and B**

d) tell( )

**Q3 Consider the following code: (Score- 2)**

template<typename T>

T maxValue(const T &val1, const T &val2)

{

if(val1>val2)

return val1;

else

return val2;

}

**Which of the following statements are correct?**

1. cout<<maxValue(1,2)
2. cout<<maxValue(1.5,2.5)
3. cout<<maxValue(‘A’, ‘B’)
4. **All of above**

**Q4.Template prefix for two parameters may be defined as: (Score -2 )**

1. Template<typename T1,typename T2>
2. Template<class T1,class T2>
3. Template<typename T1,T2>
4. **Both a and b**

**Q5. What will be the output of the line 9? (Score -2 )**

#include<fstream>

#include<iostream>

int main()

{

1 ofstream outFile;

2 outFile.open("Text1.txt");

3 char str[20[;

4 cin>>str;

5 outFile<<str;

6 ifstream inFile;

7 inFile.open("Text2.txt");

8 inFile>>str;

9 cout<<str;

10 inFile.close();

return 0;

}

1. Display updated content of file Text1
2. **Display updated content of file Text2**
3. Compiler Error
4. Display combined content of both the files

**CODING SECTION**

*PROBLEM STATEMENT-1(10 marks)*

You are given a task to maintain list of some numbers. Write a program that inserts a number before some specified number in the list. Refer sample input and output for scenario.

**Sample Input Test Case 1:**

10 // (N) size of list

1 2 3 4 5 6 7 8 9 10 // elements of the list

8 // number before which the new number to be entered

0 //new number to be inserted

**Sample Output Test Case 1:**

1 2 3 4 5 6 7 0 8 9 10 //contents of updated list

**Sample Input Test Case 2:**

5 // (N) size of list

1 3 5 6 8 // elements of the list

3 // number before which the new number to be

2 //new number to be inserted

**Sample Output Test Case 2:**

1 2 3 5 6 8 //contents of updated list

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

**Explanation:**

**Sample Input:**

First line denotes size of list

Second line denote element to be entered

Third line denote a number before which a new number is to be inserted

Fourth line denotes new number to be inserted

**Sample Output:**

Sample output display contents of updated list

**Head:**

#include <iostream>

#include <algorithm>

#include <list>

using namespace std;

int main() {

list<int> L;

**Tail:**

for(itr=L.begin();itr!=L.end();itr++)

cout<<\*itr<<"";

return 0;

}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testcase0**  **(sample)**  **(Score-0)**  **Input**  10  1 2 3 4 5 6 7 8 9 10  8  0  **Output**  1 2 3 4 5 6 7 0 8 9 10 | **Testcase1**  **(sample)**  **(Score-0)**  **Input**  5  1 3 5 6 8  3  2  **Output**  1 2 3 5 6 8 | **Testcase2**  **(Score-2)**  **Input**  4  1 5 7 8  5  6  **Output**  1 6 5 7 8 | **Testcase3**  **(Score-2)**  **Input**  2  8 9  8  3  **Output**  3 8 9 | **Testcase4**  **(Score-2)**  **Input**  1  2  2  3  **Output**  3 2 | **Testcase5**  **(Score-2)**  **Input**  6  5 6 7 8 9 0  0  2  **Output**  5 6 7 8 9 2 0 | **Testcase6**  **(Score-2)**  **Input**  3  4 5 6  6  4  **Output**  4 5 4 6 |