**MCQ SECTION  
Q1. Choose the appropriate option, (Score- 2)**

#include<iostram>

#include<deque>

using namespace std;

int main ()

{

deque mydeque;

mydeque.push\_back(100);

mydeque.push\_front(100);

mydeque.push\_back(200);

mydeque.push\_back(300);

mydeque.push\_front(100);

mydeque.push\_front(200);

mydeque.push\_front(300);

int sum=0;

while (!mydeque.empty())

{

sum+=mydeque.back();

mydeque.pop\_back();

}

cout << sum <<endl;

return 0;

}

1. 1000
2. 1100
3. 1200
4. **1300**

**Q2.Fill the code to get the output. (Score 2)**

#include <iostream>

using namespace std;

template<class T>

//Fill your code here

{

return s;

}

int main()

{

string alpha("computer");

cout<< square(alpha) <<endl;

}

1. None of the Above
2. T square (string s)
3. String square(T s)
4. **T square(T s)**

#### Q3Which is among following is used to Open a file for output and move the read/write control to the end of the file? (Score 2)

#### ios::end

#### ios::ate

#### ios::at

#### ios::move

**Q4.What is meant by exception specification? (Score 2)**

a**) A function is limited to throwing only a specified list of exceptions.**

b) A catch can catch all types of exceptions.

c) A function can throw any type of exceptions.

d) none of the mentioned

.

**Q5. What is output of following code?**

#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

**CODING SECTION**

*PROBLEM STATEMENT-1(10 marks)*

There are some books in book store having some number written on it. A customer wants to buy a book with even number written on it. So help the customer to find the number of books present in bookstore on which even number is written.

**Sample Input Test Case 1:**

4 // (N) total number of books

3 4 7 8 // numbers written on each book

**Sample Output Test Case 1:**

2 //count of even numbered books

**Sample Input Test Case 2:**

7 // (N) total number of books

1 2 3 4 5 6 7 // numbers written on each book

**Sample Output Test Case 2:**

3 // count of even numbered books

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

**Explanation:**

**Sample Input:**

First line denotes total number of books present in store

Second line denotes number written on each book

**Sample Output:**

First line denotes count of books having even number written on it

**Head:**

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

bool isEven(int i)

{

return ((i%2)==0);

}

int main()

{

vector<int> v;

int mycount; //count of even numbers

**Tail:**

cout<<mycount;

return 0;

}

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