Experiment:1.1

Name: ABHAY KUMAR **UID:20BCS9222**

Batch/Sec: 616- A **Subject: Competitive Coding**

BRANCH :- CSE

Q1)

Aim: Arrays: To implement the concept of Dynamic Array. Objective: -

To learn about Array Data Structure

To learn different approaches to reverse the elements in

Array. Problem statement:

Example

Print 4 3 2 1. Each integer is separated by one space.

Input Format

The first line contains an integer (the size of our array).
The second line contains space-separated integers that describe array 's elements.

Output Format

Print the elements of array in reverse order as a single line of space-separated numbers.

Sample Input

4

1 4 3 2

Sample Output

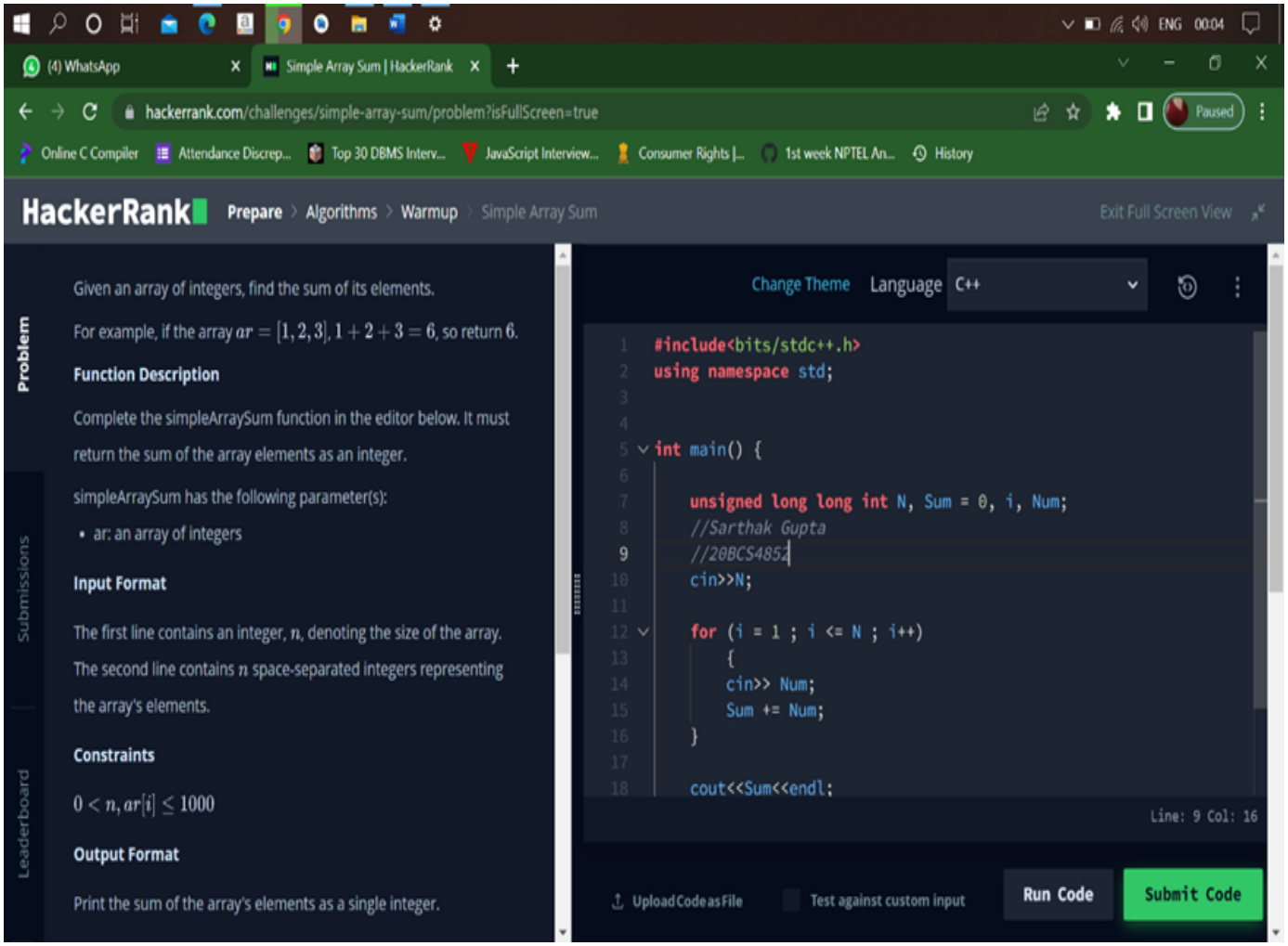
2 3 4 1

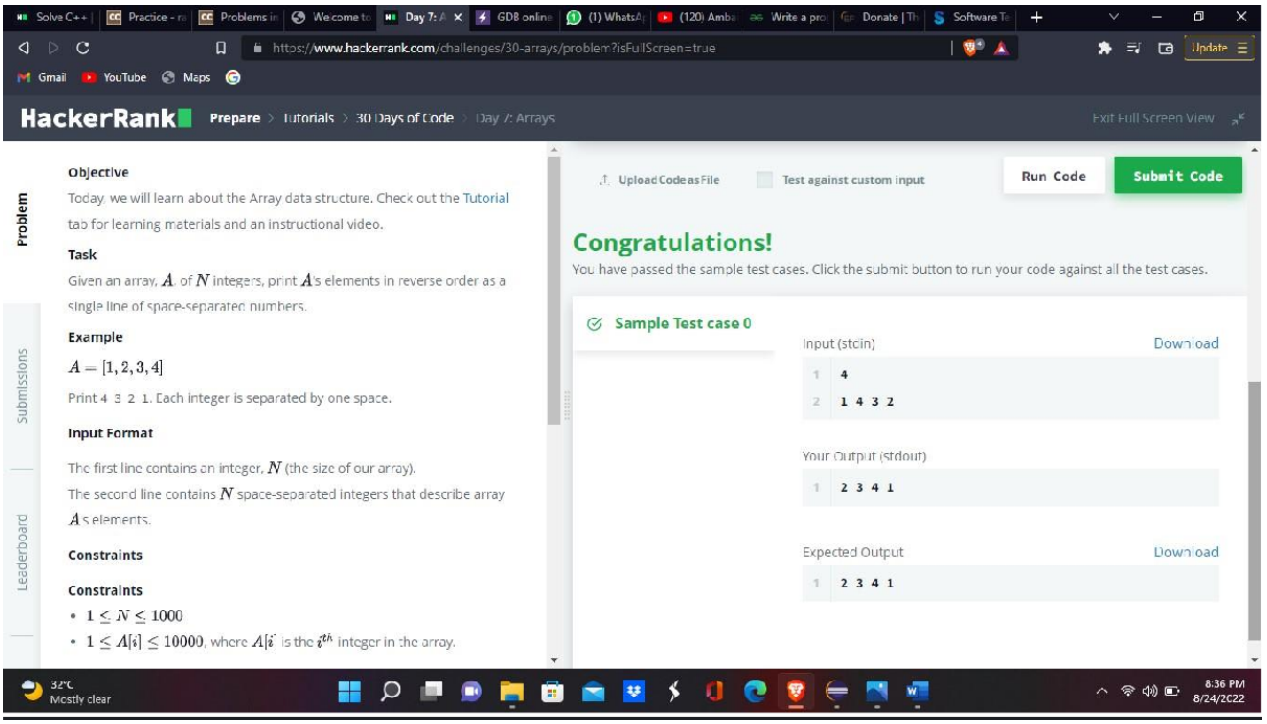
CODE:

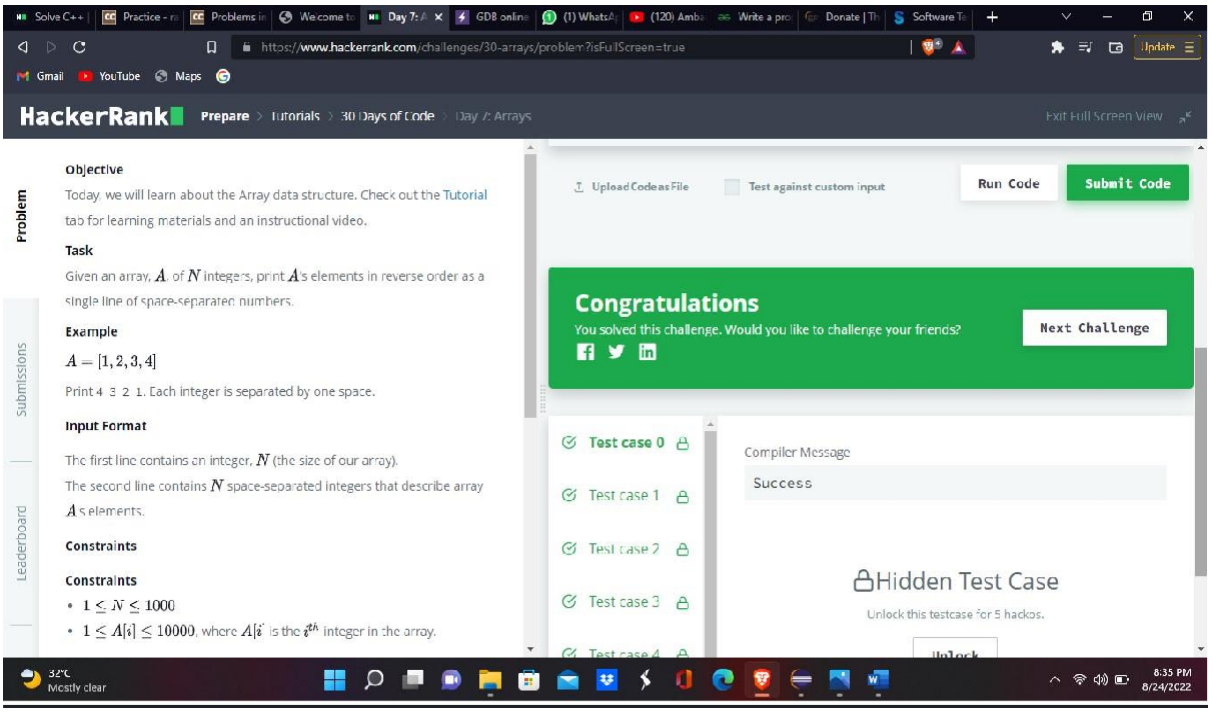
```
#include <iostream>
```

```
using namespace std; int main(){
                    int n;
cin>>n;             int a[n]; for(int
i=0;i<=n;i++){ cin>>a[i];
}for(int i=n- 1;i>=0;i--){
cout<<a[i]<<" ";
        }
}
```

Output:







Q2)

Given an array of integers, find the sum of its

elements. For example, if the array , , so return .

Function Description

Complete the *simpleArraySum* function in the editor below. It must return the sum of the array elements as an integer.

simpleArraySum has the following parameter(s):

- ***ar*: an array of**

integers Input Format

The first line contains an integer, , denoting the size of the array.

The second line contains space-separated integers representing the array's elements.

Constraints

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

6
1 2 3 4 10 11
Sample Output

31

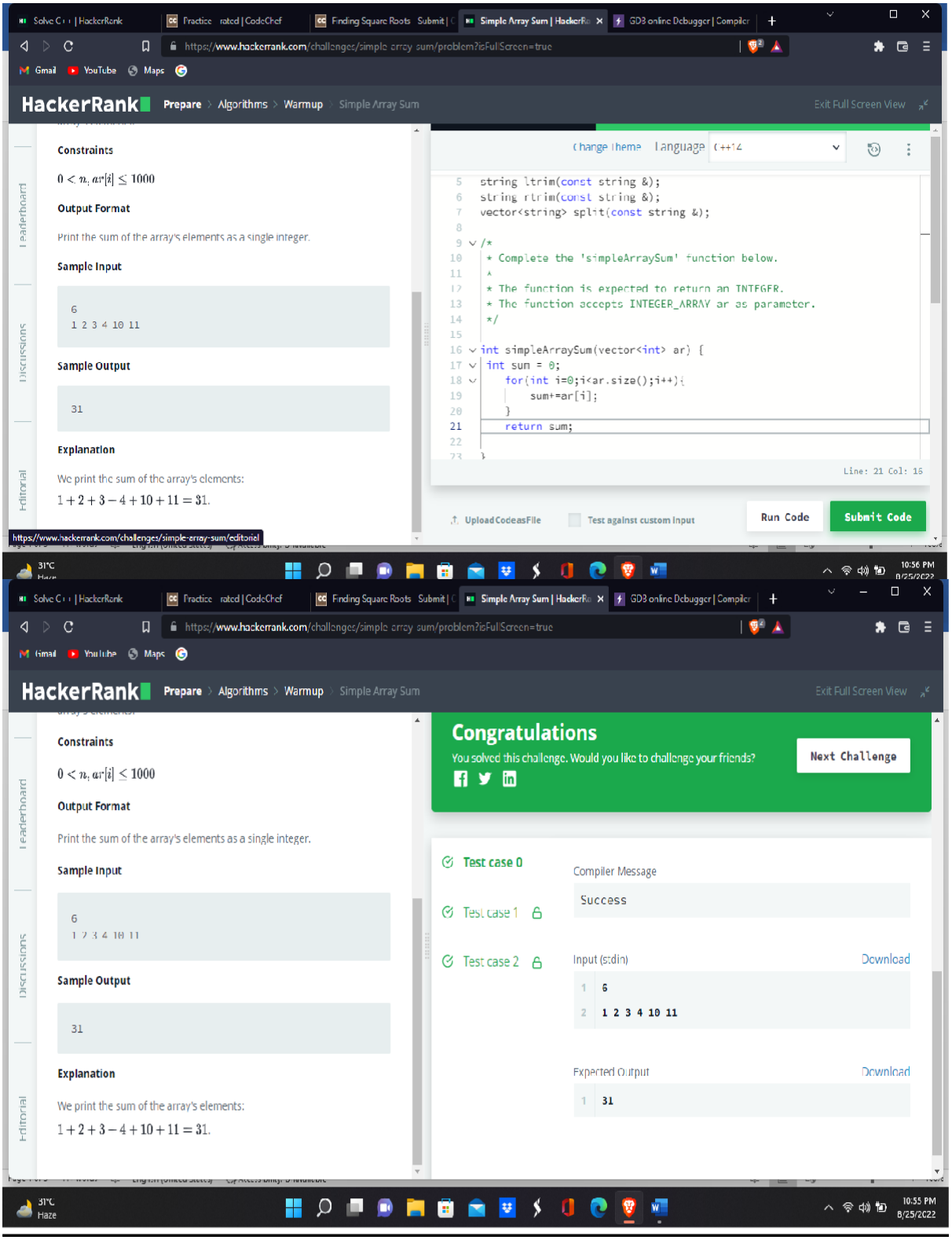
CODE:

```
#include <bits/stdc++.h> using

namespace std;

string ltrim(const string &); string
rtrim(const string &);
vector<string> split(const string &);

/*
 *  Complete the 'simpleArraySum' function below.
 *
 *  The function is expected to return an INTEGER.
 *  The function accepts INTEGER_ARRAY ar as parameter.
 */

int simpleArraySum(vector<int> ar) { int sum =
 0;
      for(int i=0;i<ar.size();i++){ sum+=ar[i];
      }
      return sum;

}

int main()
{
      ofstream fout(getenv("OUTPUT_PATH"));

      string ar_count_temp; getline(cin,
      ar_count_temp);
      int ar_count = stoi(ltrim(rtrim(ar_count_temp))); string


      ar_temp_temp;


      getline(cin, ar_temp_temp);

      vector<string> ar_temp = split(rtrim(ar_temp_temp)); vector<int>

      ar(ar_count);

      for (int i = 0; i < ar_count; i++) { int ar_item =
            stoi(ar_temp[i]);

            ar[i] = ar_item;
      }
```

```
    int result = simpleArraySum(ar); fout <<

    result << "\n"; fout.close();

    return 0;
}

string ltrim(const string &str) { string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) { string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace)))
.base(),
        s.end()
    );

    return s;
}

vector<string> split(const string &str) { vector<string>
    tokens;

    string::size_type start = 0;
    string::size_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));

        start = end + 1;
    }

    tokens.push_back(str.substr(start));
```

return tokens;
}

OUTPUT:



Learning Outcomes:

1) **Learn about C++.**
2) **Learn about Array and how to Reverse Array.**
3) **Learn About elements of array.**
4) **How to sum elements of array.**

`Learn about functions and vector`