19CSE201

LABSHEET 7

BY

JAIFIN B ALOOR

AM.SC.U4CYS23022

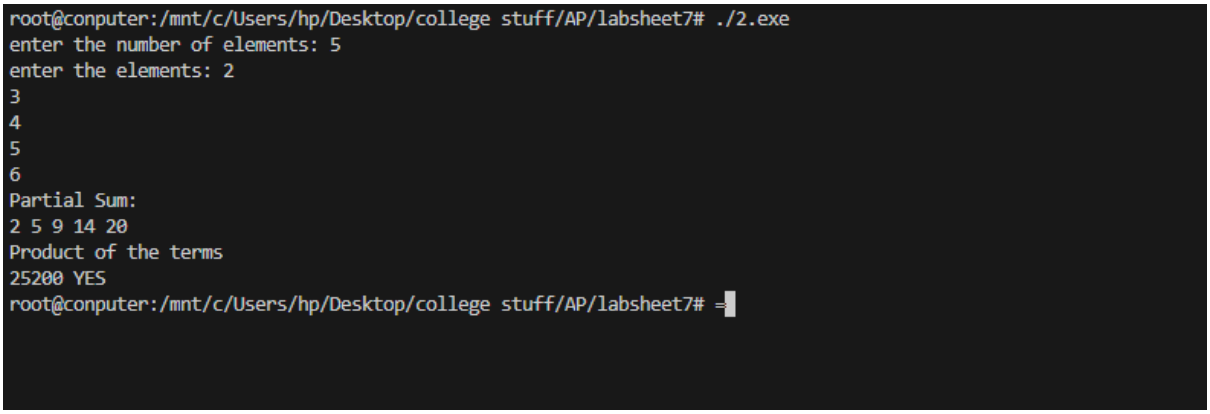
1. You are given a list of student records, where each record consists of the student's name, age, and score in a particular subject. Your task is to process these records and perform some operations based on certain criteria. Complete the given C++ function with the following signature: tuple processStudentRecords(const vector<tuple>& records);

```cpp
#include <iostream>
#include <vector>
#include <tuple>
using namespace std;

tuple<int, double> processStudentRecords(const vector<tuple<string,int, double>>& records){

    int age=0;
    int size=records.size();
    double score=0;
    for(int i=0;i<size;i++){
        tuple<string,int,double>record=records[i];
        age+=get<1>(record);
        score=(score < get<2>(record))?get<2>(record):score;
    }
    return make_tuple(age/size,score);
}

int main() {
    int n;
    cout<<"enter the number of student records: ";
    cin>>n;
    vector<tuple<string,int,double>> records;
    for(int i=0;i<n;i++){
        string name;
        int age;
        double score;
        cout<<"enter the students name, age and score: ";
        cin>>name>>age>>score;
        tuple<string,int,double> t (name,age,score);
        records.push_back(t);
    }
    tuple <int,double> result=processStudentRecords(records);
    cout<<"average age: ";
    cout<<get<0>(result)<<endl;
    cout<<"highest score: ";
    cout<<get<1>(result)<<endl;

    return 0;
}
```

```
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./1.exe
enter the number of student records: 3
enter the students name, age and score: alice 20 89
enter the students name, age and score: bob 22 90
enter the students name, age and score: charlie 22 78
average age: 21
highest score: 90
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7#
```

2. Given N positive integers, write a program to calculate the product of partial sums of the integers. Also find whether the product of partial sums is divisible by both the minimum and maximum of the given integers. See sample test for more clarity.

```cpp
#include <iostream>
#include <vector>
#include <numeric>
using namespace std;

long mul(int a, int b){
    return a*b;
}

void isDivisible(vector<int>& v, long p){
    int min=v[0],max=v[0];
    for(int i=1;i<v.size();i++){
        min=(min>v[i])?v[i]:min;
        max=(max<v[i])?v[i]:max;
    }
    if(p%min==0 && p%max==0){
        cout<<" YES\n";
    } else {
        cout<<" NO\n";
    }
}

int main() {
    int n;
    cout<<"enter the number of elements: ";
    cin>>n;
    vector<int> v;

    cout<<"enter the elements: ";
    for(int i=0;i<n;i++){
        int j;
        cin>>j;
        v.push_back(j);
    }

    vector<int> v1(v.size());
    partial_sum(v.begin(),v.end(),v1.begin());

    cout<<"Partial Sum: \n";
    for(int i=0;i<n;i++){
        cout<<v1[i]<<" ";
    }
    cout<<"\nProduct of the terms\n";
    long p = accumulate(v1.begin(),v1.end(),1,mul);
    cout<<p;
    isDivisible(v,p);
    return 0;
}
```


root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./2.exe
enter the number of elements: 5
enter the elements: 2
3
4
5
6
Partial Sum:
2 5 9 14 20
Product of the terms
25200 YES
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7#

3. Given a string a S consisting of only (and). Find whether SS is a valid parenthesis string.

```cpp
#include <iostream>
#include <vector>
#include <stack>
using namespace std;

bool isValid(string s){
    s.append(s);
    stack<char> stack;
    for(int i=0;i<s.length();i++){
        if(s[i]=='('){
            stack.push(s[i]);
        } else {
            if(stack.empty()) return false;
            char t= stack.top();
            if((s[i]==')' && t!='('){
                return false;
            } else {
                stack.pop();
            }
}
}
return stack.empty();
}
int main() {
string s1="()(())";
string s2="(()()";
string s3="))((";
cout<<isValid(s1)<<endl;
cout<<isValid(s2)<<endl;
cout<<isValid(s3)<<endl;
return 0;
}
```

```
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# g++ -o 3.exe 3.cpp
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./3.exe
1
0
0
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7#
```

4. Your best friend has a very interesting necklace with n pearls. On each of the pearls of the necklace there is an integer. However, your friend wants to modify the necklace a bit and asks you for help. She wants to move the first pearl k spots to the left (and do so with all other pearls). For example: if the necklace was originally 1,5,3,4,2 and k=2, now it becomes 3,4,2,1,5. Solve this problem using a queue.

```cpp
#include <iostream>
#include <queue>
using namespace std;
void rotate(queue<int>& necklace, int k){
for(int i=0;i<k;i++){
int pearl=necklace.front();
necklace.pop();
necklace.push(pearl);
}
}
int main() {
int n,k;
cin>>n>>k;
queue<int> necklace;
for(int i=0;i<n;i++){
        int tem;
        cin>>tem;
        necklace.push(tem);
    }

    rotate(necklace,k);
    while(!necklace.empty()){
        cout<<necklace.front()<<" ";
        necklace.pop();
    }
    return 0;
}
```

```
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./4.exe
6 5
1 2 3 4 5 6
6 1 2 3 4 5 root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7#
```

5. Given a List, the task is to delete a range of values from this List using Iterator.

```cpp
#include <iostream>
#include <list>
using namespace std;

void deleteList(list<int>& list, int start, int end){
    auto it_s=list.begin();
    for(int i=1;i<start;i++){
        it_s++;
    }

    auto it_e=list.begin();
    for(int i=1;i<end;i++){
        it_e++;
    }

    list.erase(it_s,it_e);
}

int main() {
    int n, s, e;
    cin>>n>>s>>e;
    list<int> list;
    for(int i=0;i<n;i++){
        int tem;
        cin>>tem;
list.push_back(tem);
}
deleteList(list,s,e);
for(auto it=list.begin();it!=list.end();it++){
cout<<*it<<" " <<endl;
}
return 0;
}
```

```
^[[Aroot@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./5.exe
9 3 8
10 20 30 40 50 60 70 80 90
10
20
80
90
```

6. Write a menu driven C++ program using to implement a phone directory.

```
#include <iostream>
#include <unordered_map>
using namespace std;
void addContact(unordered_map<string,int>& dictionary){
string name;
int number;
cout<<"Enter Name: ";
cin>>name;
cout<<"Enter Number: ";
cin>>number;
dictionary[name]=number;
cout<<"Contact added to unordered map\n";
}
void search(unordered_map<string,int>& dictionary){
cout<<"Enter Name: ";
string name;
cin>>name;
auto it= dictionary.find(name);
if(it!=dictionary.end()){
cout<<"Name: "<<name<<" Number: "<<it->second;
}
else {
cout<<"Name not found\n";
}
}
void deleteContact(unordered_map<string,int>& dictionary){
    cout<<"Enter Name: ";
    string name;
    cin>>name;
    dictionary.erase(name);
    cout<<"Contact deleted\n";
}
```

```cpp
int main() {
    unordered_map<string,int> dictionary;
    while(1){
        cout<<"1. Add contacts to directory\n";
        cout<<"2. Search for a contact\n";
        cout<<"3. Delete a contact\n";
        cout<<"4. Exit\n";
        int q;
        cin>>q;
        if(q==1) addContact(dictionary);
        else if(q==2) search(dictionary);
        else if(q==3) deleteContact(dictionary);
        else if(q==4) break;
        else throw "Invalid Input";
    }
    return 0;
}
```

```
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./6.exe
1. Add contacts to directory
2. Search for a contact
3. Delete a contact
4. Exit
2
Enter Name: jaifin
Name not found
1. Add contacts to directory
2. Search for a contact
3. Delete a contact
4. Exit
```

7. Write a C++ program to split a given sentence into words. Words are separated by whitespace.

```cpp
#include <iostream>
#include <vector>
using namespace std;

vector<string> split_sentence(string& s){
    vector<string> words;
    string word="";
    for(int i=0;i<s.length();i++){
        if(s[i]!=' '){
            word+=s[i];
        }
        else {
            words.push_back(word);
            word="";
        }
    }
    words.push_back(word);
return words;
}
int main() {
string s="Geeks for Geeks";
vector<string> words= split_sentence(s);
for(int i=0;i<words.size();i++){
cout<<words[i]<<endl;
}
return 0;
}
```

```
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# g++ -o 7.exe 7.cpp
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7# ./7.exe
Geeks
for
Geeks
root@conputer:/mnt/c/Users/hp/Desktop/college stuff/AP/labsheet7#
```