



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Фундаментальные науки» (ФН)
КАФЕДРА «Прикладная математика» (ФН2)

Лабораторная работа от 01.11.2024
ПО КУРСУ
«Информатика»

Студент ФН2-12Б
(Группа)

Шинкарев М.В.
(И. О. Фамилия)

Преподаватель:

Мартиросян В.В.
(И.О. Фамилия)

ЛР4

Определите структуру «Пассажир» и поместите в нее следующую информацию:

ФИО пассажира, массив элементов структуры «багаж» (поля: название предмета багажа, вес предмета в кг), при необходимости можно использовать дополнительные поля (например, число элементов в массиве багажа). В программе задайте несколько пассажиров (массив переменных структурного типа) и выполните следующие действия.

- 21) Определите число пассажиров, вес багажа которых превышает 30 кг.
- 22) Определите, имеется ли пассажир, багаж которого состоит из одной вещи весом более чем в 20 кг.
- 23) Определите средний вес багажа для пассажира.
- 24) Определите количество пассажиров, вес багажа которых превосходит средний.
- 25) Определите количество пассажиров, имеющих более трех вещей в багаже.
- 26) Определите средний вес одного предмета багажа для каждого пассажира.
- 27) Определите пассажира с максимальным весом багажа.

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
struct luggage{
    double mass=0; //масса вещи
    string thing="jkjkgjhg";//название вещи
};
struct passanger{
    string fullName="noname";//имя пассажира
    int numofthings=1;//количество вещей у пассажира
    vector<luggage> things;//вектор вещей
};
int main(){
    int n;//вводим колво пассажиров
    cin>>n;
    cin.ignore();
    vector<passanger> listofPassengers(n);//создаем вектор пассажиров
    for(int i=0;i<n;i++){
        getline(cin,listofPassengers[i].fullName);//вводим имена
        cin>>listofPassengers[i].numofthings;//вводим колво вещей пассажира
        cin.ignore();

        listofPassengers[i].things.resize(listofPassengers[i].numofthings);//перепред
        еляем размер вектора вещей
        for(int j=0;j<listofPassengers[i].numofthings;j++){
            cin>>listofPassengers[i].things[j].mass;//вводим массу вещи
            cin.ignore();
            getline(cin,listofPassengers[i].things[j].thing);//вводим название
            вещи
        }
    }
}
```

```

        //cout<<listofPassengers[0].fullName<<"
"<<listofPassengers[0].numofthings<<" "<<listofPassengers[0].things[0].mass<<"
"<<listofPassengers[0].things[0].thing;
        for(int i=0;i<n;i++){// здесь выводим всех пассажиров и их данные
            cout<<listofPassengers[i].fullName<<endl;
            cout<<"Passanger has "<<listofPassengers[i].numofthings<<endl;
            for(int j=0;j<listofPassengers[i].numofthings;j++){
                cout<<listofPassengers[i].things[j].mass<<"
"<<listofPassengers[i].things[j].thing<<" ";
            }
            cout<<endl;
        }

//задание 1 Определите число пассажиров, вес багажа которых превышает 30
кг.
//void countPEXtraLuggageMass(){
    double massOneLuggagePassenger=0;
    int countPEXtraLuggageMass=0;
    for(int i=0; i<n; i++){
        for(int j=0; j<listofPassengers[i].numofthings;j++){
            massOneLuggagePassenger+=listofPassengers[i].things[j].mass;
        }
        if(massOneLuggagePassenger>30){
            countPEXtraLuggageMass+=1;
        }
        massOneLuggagePassenger=0;
    }
    cout<<"Имеется "<<countPEXtraLuggageMass<<" пассажиров с весом багажа
более 30 кг"<<endl;
    //}

//задание 2 Определите, имеется ли пассажир, багаж которого состоит из
одной вещи весом более чем в 20 кг.
//void countPwithHeavyLuggage(){
    int countPwithHeavyLuggage=0;
    for(int i=0; i<n; i++){
        for(int j=0; j<listofPassengers[i].numofthings;j++){
            if(listofPassengers[i].things[j].mass>20){
                countPwithHeavyLuggage+=1;
                break;
            }
        }
    }
    if(countPwithHeavyLuggage>0){
        cout<<"Имеется "<<countPwithHeavyLuggage<<" пассажиров с очень
тяжелой вещью, весом более 20 кг"<<endl;
    }
}

```

```

//}

//задание 3 Определите средний вес багажа для пассажира.
//int GetAverageMassLuggage(){
    double massLuggagePassengers=0;
    double averageMassLuggage;
    for(int i=0; i<n; i++){
        for(int j=0; j<listofPassengers[i].numofthings;j++){
            massLuggagePassengers+=listofPassengers[i].things[j].mass;
        }
    }
    averageMassLuggage=massLuggagePassengers/n;
    //return averageMassLuggage;
//}

//void printAverageMassLuggage(int averageMassLuggage){
    cout<<"Средний вес багажа "<<averageMassLuggage<<endl;
//}

//задание 4 Определите количество пассажиров, вес багажа которых
превосходит средний.
//void countPEXtraLuggageMass(){
    massOneLuggagePassenger=0;
    int countPOverAverageLuggageMass=0;
    for(int i=0; i<n; i++){
        for(int j=0; j<listofPassengers[i].numofthings;j++){
            massOneLuggagePassenger+=listofPassengers[i].things[j].mass;
        }
        if(massOneLuggagePassenger>averageMassLuggage){
            countPOverAverageLuggageMass+=1;
        }
        massOneLuggagePassenger=0;
    }
    cout<<"Имеется "<<countPOverAverageLuggageMass<<" пассажиров, вес
багажа которых превосходит средний"<<endl;
//}

//задание 5 Определите количество пассажиров, имеющих более трех вещей в
багаже.
//void countPEXtraNumofthings(){
    int countPEXtraNumofthings=0;
    for(int i=0; i<n; i++){
        if(listofPassengers[i].numofthings>3){
            countPEXtraNumofthings+=1;
        }
    }
    cout<<"Имеется "<<countPEXtraNumofthings<<" пассажиров, имеющих более
трех вещей в багаже"<<endl;

```

```

    //}

    //задание 6 Определите средний вес одного предмета багажа для каждого
    пассажира.
    //void countPEXtraNumofthings(){
        double PMassofluggage=0;
        int Allnumofthings=0;
        for(int i=0; i<n; i++){
            for(int j=0; j<listofPassengers[i].numofthings; j++){
                PMassofluggage+=listofPassengers[i].things[j].mass;
            }
            Allnumofthings+=listofPassengers[i].numofthings;
        }
        cout<<(PMassofluggage/Allnumofthings)<<" средний вес багажа"<<endl;
    //}

    //задание 7 Определите пассажира с максимальным весом багажа.
    //void PwithMaxLuggage(){
        double MaxMass=0;
        double massluggage=0;
        string name=listofPassengers[0].fullName;
        for(int i=0; i<n; i++){
            for(int j=0; j<listofPassengers[i].numofthings; j++){
                massluggage+=listofPassengers[i].things[j].mass;
            }
            if(MaxMass<massluggage){
                MaxMass=massluggage;
                name=listofPassengers[i].fullName;
            }
            massluggage=0;
        }
        cout<<"Пассажир "<<name<<" "<<" имеет максимальный вес багажа
"<<MaxMass<<endl;
    //}
    system("pause");
    return 0;
}
//ВВОД
/*5
shinkarev mark
3
5
sumka
35
dinosaur
1
phone

```

```
Chel
2
0.3
potatoes
0.7
portal gun
chel2
4
3
ohapka drow
0.5
ukrop
2.5
cat tail
29
kirpichi
chel3
1
3
without tail
Swarz Negger
5
4
Chel
100
chel2
20
chel3
5
machine gun
10
patronyi*/

/*ВЫВОД
shinkarev mark
Passanger has 3
5 sumka 35 dinosaur 1 phone
Chel
Passanger has 2
0.3 potatoes 0.7 portal gun
chel2
Passanger has 4
3 ohapka drow 0.5 ukrop 2.5 cat tail 29 kirpichi
chel3
Passanger has 1
3 without tail
```

Swarz Negger

Passanger has 5

4 Chel 100 chel2 20 chel3 5 machine gun 10 patronyi

Имеется 3 пассажиров с весом багажа более 30 кг

Имеется 3 пассажиров с очень тяжелой вещью, весом более 20 кг

Средний вес багажа 43.8

Имеется 1 пассажиров, вес багажа которых превосходит средний

Имеется 2 пассажиров, имеющих более трех вещей в багаже

14.6 средний вес багажа

Пассажир Swarz Negger имеет максимальный вес багажа 139*/

№1

Формат ввода

Программа получает на вход шесть чисел в следующем порядке: x_1, y_1 (координаты юго-западного угла плота), x_2, y_2 (координаты северо-восточного угла плота), x, y (координаты пловца). Все числа целые и по модулю не превосходят 100. Гарантируется, что $x_1 < x_2, y_1 < y_2, x \neq x_1, x \neq x_2, y \neq y_1, y \neq y_2$, координаты пловца находятся вне плота.

Формат вывода

Если пловцу следует плыть к северной стороне плота, программа должна вывести символ "N", к южной — символ "S", к западной — символ "W", к восточной — символ "E". Если пловцу следует плыть к углу плота, нужно вывести одну из следующих строк: "NW", "NE", "SW", "SE".

Пример

Ввод

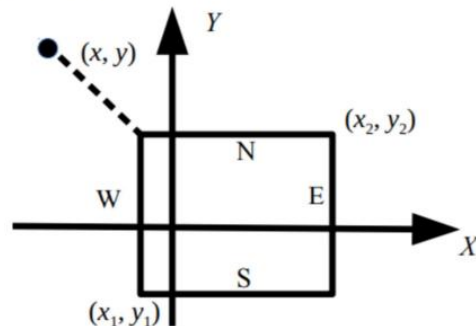
```
-1
-2
5
3
-4
6
```

Вывод

NW

Посередине озера плавает плот, имеющий форму прямоугольника. Стороны плота направлены вдоль параллелей и меридианов. Введём систему координат, в которой ось OX направлена на восток, а ось OY — на север. Пусть юго-западный угол плота имеет координаты (x_1, y_1) , северо-восточный угол — координаты (x_2, y_2) .

Пловец находится в точке с координатами (x, y) . Определите, к какой стороне плота (северной, южной, западной или восточной) или к какому углу плота (северо-западному, северо-восточному, юго-западному, юго-восточному) пловцу нужно плыть, чтобы как можно скорее добраться до плота.



```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    // Ввод координат плота и пловца
    int x1, y1, x2, y2, x, y;
    cin >> x1 >> y1 >> x2 >> y2 >> x >> y;

    // Проверка, находится ли пловец внутри плота
    if (x1 <= x && x <= x2 && y1 <= y && y <= y2) {
        cout << "Пловец находится внутри плота!" << endl;
        return 0;
    }
}
```



```

// Определение направления движения
if (x < x1) {
    if (y < y1) {
        cout << "SW" << endl; // Юго-западный угол
    } else if (y > y2) {
        cout << "NW" << endl; // Северо-западный угол
    } else {
        cout << "W" << endl; // Западная сторона
    }
} else if (x > x2) {
    if (y < y1) {
        cout << "SE" << endl; // Юго-восточный угол
    } else if (y > y2) {
        cout << "NE" << endl; // Северо-восточный угол
    } else {
        cout << "E" << endl; // Восточная сторона
    }
} else {
    if (y < y1) {
        cout << "S" << endl; // Южная сторона
    } else {
        cout << "N" << endl; // Северная сторона
    }
}

return 0;
}

```

Пример

Ввод

6
2
7
3

Вывод

3 4

Примечания

В примере из условия в шкафу лежит $A = 6$ синих маек и $B = 2$ красных маек. Если взять 3 майки, то среди них обязательно найдётся синяя. В другом ящике лежит $C = 7$ пар синих носков и $D = 3$ пары красных носков. Если взять 4 пары, то среди них обязательно будет пара синих носков. Поэтому если взять вслепую 3 майки и 4 пары носков, то среди них обязательно найдётся одноцветный (синий) комплект из майки и носков.

Как известно, осенью и зимой светает поздно, и так хочется утром ещё хоть немного поспать, а не идти в школу! Некоторые школьники готовы даже одеваться, не открывая глаз, лишь бы отложить момент пробуждения. Вот и Саша решил, что майку и носки он вполне может вытащить из шкафа на ощупь с закрытыми глазами и только потом включить свет и одеться. В шкафу у Саши есть два ящика. В одном из них лежит A синих и B красных маек, в другом — C синих и D красных пар носков. Саша хочет, чтобы и майка, и носки были одного цвета. Он вслепую вытаскивает M маек и N пар носков. В первое же утро Саша задумался, какое минимальное суммарное количество предметов одежды ($M + N$) он должен вытащить, чтобы среди них гарантированно оказались майка и носки одного цвета. Какого именно цвета окажутся предметы одежды, для Саши совершенно неважно.

Формат ввода

На вход программе подаются четыре целых неотрицательных числа A, B, C, D , записанных в отдельных строках: A — количество синих маек, B — количество красных маек, C — количество синих носков, D — количество красных носков. Все числа не превосходят 10^9 . Гарантируется, что в шкафу есть одноцветный комплект из майки и носков.

Формат вывода

Программа должна вывести два числа: количество маек M и количество пар носков N , которые должен взять Саша. Необходимо, чтобы среди M маек и N пар носков обязательно нашлась одноцветная пара, при этом сумма $M + N$ должна быть минимальной.

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

int main() {
    int a,b,c,d,n,m;
    cin >> a >> b >> c >> d;
    vector<pair<int,int>> var;
    if (a != 0 && c != 0) {
        if (b != 0 && d != 0) {
            var.emplace_back(b+1, d+1);
            var.emplace_back(max(a,b)+1, 1);
            var.emplace_back(1, max(c,d)+1);
        }
        else if (b != 0 && d == 0){
```

```

        var.emplace_back(b+1, 1);
    }
    else if (b == 0 && d != 0){
        var.emplace_back(1, d+1);
    }
    else{
        var.emplace_back(1, 1);
    }
}
if (b != 0 && d != 0) {
    if (a != 0 && c != 0) var.emplace_back(a+1, c+1);
    else if (a != 0 && c == 0){
        var.emplace_back(a+1, 1);
    }
    else if (a == 0 && c != 0){
        var.emplace_back(1, c+1);
    }
    else{
        var.emplace_back(1, 1);
    }
}

m=var[0].first;
n=var[0].second;
for(auto x : var){
    if(x.first+x.second<n+m){
        m=x.first;
        n=x.second;
    }
}
cout << m << " " << n << endl;
return 0;
}

```

Я создаю вектор пар для перебора всех возможных случаев, а затем пробегаюсь по ним