#include <iostream>

#include <string>

#include <vector>

#include <map>

using namespace std;

int main(){

string s = "abhjgdf";

for (char c : s){

cout <<c <<",";

}

return 0;

}

#include <iostream>

#include <string>

#include <vector>

#include <map>

using namespace std;

int main(){

vector < int> nums={1,5,2,3,5};

for (int c : nums){

cout <<c <<",";

}

return 0;

}

TWO FIVES

#include <iostream>

#include <string>

#include <vector>

#include <map>

using namespace std;

int main(){

vector<int>nums ={1,5,3,4,5};

int quantity =0;

for (auto x: nums){

if (x==5) {

quantity +=1;

}

}

cout <<"There are "<<quantity <<"fives";

return 0;

}

#include <iostream>

using namespace std;

int main() {

vector <string> w = {"a","b","c"};

vector <string> v;

v=w;

v[0]="d";

cout <<w[0]<<" "<<w[1]<<endl;

cout <<v[0]<<" "<<v[1]<<endl;

return 0;

}

Home work N1

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

/\* alpha beta gammma

beta alpha gamma

alpha belta alpha

\*/

using namespace std;

int main(){

string a, b, c;

cin >> a >> b>>c;

if (a<=b && a<=c){

cout<< a;

}else if (b <= a && b <= c){

cout<<b;

}else{

cout <<c;

}

return 0;

}

Home Work 2

#include <cmath>

#include <iostream>

using namespace std ;

int main(){

double a, b, c, D, x1, x2;

cin >> a >> b >> c;

D = b \* b -4 \* a \* c;

if((a == 0 && c == 0) || (b == 0 && c == 0)){

cout << 0 ;

}else if (a == 0) {

cout << - ( c / b);

}else if(b == 0){

cout <<" " ;

}else if (c == 0){

cout << 0 << " "<< -(b / a);

}else if(D < 0){

cout <<" " ;

}else if(D == 0){

x1=((-1 \* b)+sqrt(D))/(2 \* a);

cout << x1 ;

}else if (D > 0){

x1=((-1 \* b)+sqrt(D))/(2 \* a);

x2=((-1 \* b)-sqrt(D))/(2 \* a);

cout << x1 <<" "<< x2;

}

return 0 ;

}

IF statement:

#include <cmath>

#include <iostream>

using namespace std ;

int main(){

int a,b;

cin >> a>> b;

if(a == b){

cout<< "equal" <<endl;

cout <<a;

}

else{

cout <<" not equal" << endl;

cout <<a <<" "<<b;

}

return 0 ;

}

#2

#include <cmath>

#include <iostream>

using namespace std ;

int main(){

int a = -1;

if (a >=0){

if(a>0)

cout <<"positive";

}else{

cout<< "negative";

}

return 0 ;

}

Practice Programming Assignment: Деление

#include <iostream>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

if (b == 0) {

cout << "Impossible" << endl;

} else {

cout << a / b << endl;

}

return 0;

}

## Practice Programming Assignment: Расчёт стоимости товара

#include <iostream>

using namespace std;

int main() {

double n, a, b, x, y;

cin >> n >> a >> b >> x >> y;

if (n > b) {

n \*= (1 - y / 100);

} else if (n > a) {

n \*= (1 - x / 100);

}

cout << n;

return 0;

}

## Practice Programming Assignment: Чётные числа

#include <iostream>

using namespace std;

// Дано два целых числа ( it means int )

//int A и B

//(A <= B, A >= 1, B <= 30000). Выведите через пробел все чётные числа от

//A до B (включительно) it mean for loop. We need to do cycle .

// 1 10 2 4 6 8 10

// 2 3 2

// 9 11 10

int main(){

int a , b;

cin >> a >>b;

for ( auto i = a; i <=b; ++i){

if ( i % 2 == 0 ){

cout << i << ' ';

}

}

return 0;

}

## Programming Assignment: Второе вхождение

#include <iostream>

#include <string>

using namespace std;

int main() {

string s;

cin >> s;

int result = -2; // текущий ответ

// если f ни разу не встретится, он останется равным -2

for (int i = 0; i < s.size(); ++i) {

// если текущий символ равен f, обновим текущий ответ

if (s[i] == 'f') {

// есди до этого не было ни одного вхождения, значит, сейчас нашли первое

if (result == -2) {

result = -1; // теперь вхождение ровно одно

// до этого было ровно одно вхождение → сейчас как раз второе

} else if (result == -1) {

result = i;

break; // цикл можно завершить, потому что ответ уже найден

}

}

}

cout << result;

return 0;

}

## Programming Assignment: Наибольший общий делитель

#include <iostream>

#include <string>

using namespace std;

//В stdin даны два натуральных числа. Выведите в stdout их наибольший общий делитель.

//25 27 1

//12 16 4

//13 13 13

int main(){

int a, b;

cin >> a >> b;

//a = abs(a);

//b = abs(b);

if (a == 0 || b == 0 ){

a = max(a, b);

} else {

while ( a != b){

if (a > b){

a -= b;

} else {

b -= a;

}

}

}

cout << a;

return 0;

}

## Programming Assignment: Двоичные числа

#include <iostream>

#include <string>

using namespace std;

int main() {

string a;

int x=0;

int y=0;

cin >>a;

for (auto c: a){

if (c=='f')

{

x++;

}

if(x==2){

break;

}

y++;

}

if (x==0)

{cout << "-2" << endl;}

if (x==1)

{cout << "-1" << endl;}

if (x==2)

{cout << y << endl;}

return 0;

}

#include <iostream>

using namespace std;

int main()

{

unsigned int n;

unsigned long long factorial = 1;

cout << "Enter a positive integer: ";

cin >> n;

for(int i = 1; i <=n; ++i)

{

factorial \*= i;

}

cout << "Factorial of " << n << " = " << factorial;

return 0;

}

2222222222222222==========2222222=================222222222========222222

#include<iostream>

using namespace std;

int main(void)

{

    int a,b,c;

    cin>>a>>b;

   while (b) {

      c = a % b;

      a = b;

      b = c;

   }

cout<<a;;

system("pause");

}

Welcome WEEK 2:

#include <iostream>

using namespace std;

int Factorial(int x) {

int i, fac;

int sum = 1;

for (i=x; i>1; i=i-2) {

fac = i\*(i-1);

sum = fac \* sum;

}

return sum ;

}

int main() {

int x;

cin >> x;

if (x>=1)

cout << Factorial(x) << endl;

else if (x==0)

cout << "1";

else

cout << "-1";

return 0;

}

## Палиндром

bool IsPalindrom(string s) {

for (size\_t i = 0; i < s.size() / 2; ++i) {

if (s[i] != s[s.size() - i - 1]) {

return false;

}

}

return true;

}

#SWAP

#include <iostream>

using namespace std;

void Swap(int x, int y){

int tmp =x;

x=y;

y=tmp;

}

int main(){

int a = 1;

int b= 2;

Swap(a,b);

cout << "a ==" << a << '\n';

cout <<"b=="<<b <<'\n';

return 0;

}

Vector ######Sort

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

void Sort(vector<int> &v){

sort(begin(v), end(v));

}

int main(){

vector <int> num={3,6,1,2,0,2};

Sort(num);

for (auto x: num){

cout <<x <<" " ;

}

return 0;

}

## Practice Programming Assignment: Максимизатор

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

void UpdateIfGreater(int a, int& b){

if (a>b){

b=a;

}

}

int main(){

int a,b;

cin >>a >>b;

UpdateIfGreater(2,b);

cout <<b;

return 0;

}

## Programming Assignment: Разворот последовательности

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

void Reverse(vector<int>& v){

reverse(begin(v), end(v));

}

int main(){

vector<int> numbers = {1, 5, 3, 4, 2};

Reverse(numbers);

for (auto v: numbers){

cout <<v <<" ";

}

return 0;

}

**Контейнер vector**

Temperatures

int main(){

int n;

cin >> n;

vector <int> temperatures(n);

int sum = 0;

for (int& temperature: temperatures){

cin>> temperature;

sum +=temperature;

}

int average = sum /n;

vector <int> result\_indcies;

for (int i = 0; i<n; ++i){

if(temperatures[i]>average){

result\_indcies.push\_back(i);

}

}

cout<< result\_indcies.size() <<endl;

for (int result\_index: result\_indcies){

cout << result\_index<< " ";

}

cout <<endl;

return 0;

}

## Programming Assignment: Очередь

int main() {

int countofoper = 0; //count of operations

cin >> countofoper;

vector<int> people; //vector of queue

vector<int> worrypeople ; //vector of worried people

string s;

int num = 0;

for (int i = 0; i< countofoper; ++i) {

cin >> s;

if (s!= "WORRY\_COUNT") {

cin >> num ;

if (s == "COME") {

int currentsize = people.size() + num;

people.resize(currentsize);

}

if (s == "WORRY") {

people[num] = 1;

}

if (s == "QUET") {

people[num] = 0;

}

} else if (s == "WORRY\_COUNT") {

int w = count(begin(people), end(people), 1);

worrypeople.push\_back(w);

}

}

for (auto wp: worrypeople) {

cout << wp << endl;

}

return 0;

}

## Автобусные остановки — 1

void All\_buses(const map<string, vector<string>>& f) { //вывод всех маршрутов

    if (f.size() == 0) cout << "No buses" << endl; //если размер контейнера = 0, то напишем что ничего нет

    else { //если что-то есть

        for (const auto& i : f) { //пролистаем контейнер и напишем, для каждой пары "ключ-значение"

            cout << "Bus " << i.first << ":"; //название маршрута

            for (const auto& d : i.second) { //а также перечислим

                cout << " " << d; // название остановок

            }

            cout << endl;

        }

    }

}

void BusforStop(const string& s, const map<string, vector<string>>& f, const vector<string>& busm) { //вывод названий марштутов для остановки

    int k = 0; //определитель наличия чего-нибудь

    vector<string> v;

    for (const auto& i : f) { //листаем контейнер

        for (const auto& n : i.second) { //и в каждом массиве значений для каждого ключа

            if (n == s) { //ищем совпадение с остановкой и, если нашли,

                v.push\_back(i.first); //запишем "найденный" маршрут в вектор

                k++; //плюсанули наличие совпадения

            }

        }

    }

    if (k > 0) { //если есть совпадения

        for (const auto& w : busm) { //пролистаем общий вектор с маршрутами и для каждого маршрута

            for (auto z : v) { //будем просматривать вектор "найденных" маршрутов

                if (z == w) { //и при совпадении

                    cout << w << " "; //выводим маршрут

                } //таким образом у нас будут выведены маршруты в порядке записи командой NEW\_BUS

            }

        }

    }

    else if (k == 0) cout << "No stop"; //если не нашли совпадение, то так и напишем

    cout << endl;

}

void Stopforbus(const string& s, const map<string, vector<string>>& f, const vector<string>& busm) {

    int l = 0; //переменная для определения наличия нужного маршрута

    for (auto i : f) { //просматриваем контейнер

        if (i.first == s) { //если значение ключа совпало с названием маршрута

            for (auto m : i.second) { // (п.1) то для каждого значения этого ключа

                cout << "Stop " << m << ":"; //пишем остановку

                int l2 = 0; //переменная для определения наличия других маршрутов остановки, кроме первоначального

                vector<string> v;

                for (auto r : f) { //снова пролистываем контейнер

                    for (auto n : r.second) { //просматриваем все значения остановок

                        //и если значение остановки совпало со значением ключа m в п.1, но не равно "введенному" маршруту s

                        if (n == m && r.first != s) {

                            v.push\_back(r.first); //запишем маршруты в вектор

                            l2++;

                        }

                    }

                }

                if (l2 == 0) cout << " no interchange";

                else if (l2 > 0) {

                    for (const auto& w : busm) { //пролистаем общий вектор с маршрутами и для каждого маршрута

                        for (auto z : v) { //будем просматривать вектор "найденных" маршрутов

                            if (z == w) { //и при совпадении

                                cout << " "  << w; //выводим маршрут

                            } //таким образом у нас будут выведены маршруты в порядке записи командой NEW\_BUS

                        }

                    }

                }

                cout << endl; //с новой строки

            }

            l++; //если маршрут нашелся, то плюсуем

        }

    }

    if (l == 0) cout << "No bus" << endl; // если не нашлось маршрутов - так и пишем

}

int main()

{

    int Q;

    cin >> Q; //узнаем сколько команд будет

    map<string, vector<string>> bsmap; //создадим контейнер, ключом будет маршрут(строка), а значениями остановки (вектор строк)

    vector<string> busm;

    for (Q; Q > 0; Q--) { //делаем нужное количество циклов

        string comand, bus, stop; //заведем нужные переменные

        cin >> comand; //считываем команду

        if (comand == "NEW\_BUS") { //для этой команды

            int sc; //количество остановок

            cin >> bus >> sc; //прочитаем маршрут и количество остановок

            busm.push\_back(bus);

            for (sc; sc > 0; sc--) { //и ровно это количество раз

                cin >> stop; //считаем название новой остановки

                bsmap[bus].push\_back(stop); //и добавим к маршруту

            }

        }

        else if (comand == "BUSES\_FOR\_STOP") {

            cin >> stop;

            BusforStop(stop, bsmap, busm); //тут и ниже все происходит в функциях, которые выше

        }

        else if (comand == "STOPS\_FOR\_BUS") {

            cin >> bus;

            Stopforbus(bus, bsmap, busm);

        }

        else if (comand == "ALL\_BUSES") {

            All\_buses(bsmap);

        }

    }

    return 0;

}

i < n / 2;

165

1