

```

    }

    printf("NO\n");

    return 0;
}

```

HASHING :-

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information
				Level 1 Challenge 91

Problem Description

You are given with integers a, b, c, d, m . These represent the modular equation of a curve $y^2 \bmod m = (ax^3 + bx^2 + cx + d) \bmod m$. Also, you are provided with an array A of size N . Now, your task is to find the number of pairs in the array that satisfy the given modular equation.

If (A_i, A_j) is a pair then $A_j^2 \bmod m = (aA_i^3 + bA_i^2 + cA_i + d) \bmod m$.

Since the answer could be very large output it modulo $10^9 + 7$.

Note: A pair is counted different from some other pair if either A_i of the two pairs is different or A_j of the two pairs is different. Also for the convenience of calculations, we may count (A_i, A_i) as a valid pair if it satisfies given constraints.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^5$
- $-2 \times 10^9 \leq a, b, c, d, A_i \leq 2 \times 10^9$
- $1 \leq m \leq 2 \times 10^9$

Input Format

First line of the input contains number of test cases T .
 First line for each test case consists of 5 space-separated integers a, b, c, d, m , corresponding to modular equation given.
 Next line contains a single integer N .
 Next line contains N space-separated integers corresponding to values of array A .

Output Format

For each test case, output a single line corresponding to number of valid pairs in the array mod $10^9 + 7$.

Logical Test Cases

```

#include <bits/stdc++.h>

using namespace std;

const int md = 1E9 + 7;

map<long long, int> mp;

int main() {

    int t;

    cin >> t;

    while(t--) {

        long long a, b, c, d, m;

        cin >> a >> b >> c >> d >> m;

        int n;

        cin >> n;

        int arr[n];

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

            cin >> arr[i];

            mp[(((arr[i] * arr[i]) % m) + m) % m]++;

```

```

    }

    long long ans = 0;

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

        long long x = (((((((a * arr[i]) % m) * ((arr[i] * arr[i]) % m)) % m) + (b * ((arr[i] * arr[i]) % m) % m) + ((c * arr[i]) % m) + d) % m) + m) % m;

        if(mp.find(x) != mp.end())

            ans += mp[x];

    }

    cout << (ans % md) << '\n';

    mp.clear();

}

return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information
				Level 1 Challenge 92

Question description

The students of college *XYZ* are getting jealous of the students of college *ABC*. *ABC* managed to beat *XYZ* in all the sports and games events. The main strength of the students of *ABC* is their unity. The students of *XYZ* decide to destroy this unity. The geeks of *XYZ* prepared a special kind of perfume. Anyone who inhales this perfume becomes extremely violent. The students of *XYZ* somehow manage to spread this perfume throughout *ABC*'s campus atmosphere.

There are N boys $(1, 2, 3, \dots, N)$ and N girls $(1, 2, 3, \dots, N)$ in *ABC* college. Each boy has a crush on a single girl and each girl has a crush on a single boy. Since the perfume has been inhaled by each and every student of *ABC* college, every student decides to beat up his/her crush's crush, i.e., if boy x has a crush on girl y and girl y has a crush on boy z , x will beat z up, provided, of course, if x and z is not the same person.

The doctor of *ABC* college foresees this situation. He cannot stop so many people from beating each other up, however, he can be prepared for the worst-case patient(s). The worst-case patient(s) will be the patient(s) who get(s) beaten up by the maximum number of students. The doctor comes to you for help. He has 2 questions for you :

1. What is the number of beatings received by the worst-case patient(s) ?
2. What is the total number of pairs of students who ended up beating up each other ?

Constraints :

$1 \leq T \leq 10$
 $1 \leq N \leq 10^5$

Input :

The first line comprises of T , the number of test cases. Each test case comprises of 3 lines. The first line consists of N .

The next line consists of N space separated natural numbers between 1 and N inclusive such that the i th number denotes the the crush of boy i .

The next line consists of N space separated natural numbers between 1 and N inclusive such that the i th number denotes the the crush of girl i .

Output :

For every test case, on a new line, print two space separated integers, the answer to doctor's question 1 followed by answer to doctor's question 2.

```

#include<stdio.h>

void solve(){

int main()

{

    solve();

    int t,n,b[100010],g[100010],i;

    scanf("%d",&t);

    while(t--)

    {

        int bbeat[100010]={0},gbeat[100010]={0};

```

```

scanf("%d",&n);

for(i=1;i<=n;i++)
{
    scanf("%d",&b[i]);
}

for(i=1;i<=n;i++)
{
    scanf("%d",&g[i]);
}

for(i=1;i<=n;i++)
{
    if(g[b[i]]!=i)
    {
        bbeat[g[b[i]]]++;
    }

    if(b[g[i]]!=i)
    {
        gbeat[b[g[i]]]++;
    }
}

int max=-1;

for(i=1;i<=n;i++)
{
    if(bbeat[i]>max)
    {
        max=bbeat[i];
    }

    if(gbeat[i]>max)
    {
        max=gbeat[i];
    }
}

int count=0;

for(i=1;i<=n;i++)
{
    if(g[b[i]]!=i && g[b[g[b[i]]]]==i)
    {
        count++;
    }
}

```

```

    }

    if(b[g[i]]!=i && b[g[b[g[i]]]==i)
    {
        count++;
    }
}

printf("%d %d \n",max,count/2);

}

return(0);

printf("while(true)");

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. The page content includes a notification: "You have already solved this challenge! Though you can run the code with different logic!". Below this is a table with columns: Course, DS, Session, Hashing, Question Information, Level 1, and Challenge 93. The "Question Information" column is expanded, showing the following details:

Question description
You are given an array A of length N which is initialised with 0. You will be given Q queries of two types:
1 K : set value I at index k in array A
2 y : print the smallest index x which is greater than or equal to y and having value I . If there is no such index print -1 .
Note: Indexing is 1 based

Constraints
 $1 \leq n \leq 10^6$
 $1 \leq q \leq 5 \times 10^5$
 $1 \leq y, k \leq n$

Input Format
First line contains two integers N and Q separated by a space.
The next Q lines contain the type of query (i.e. either a 1 or a 2), then a space, then for type 1 queries integer k and for type 2 queries integer y .

Output Format
For each query type 2, print in new line, the smallest index x which is greater than or equal to y and having value I . If there is no such index print -1 .

Explanation for Test case 1
For first query: 2 3, there is no index greater than or equal index 3, having value -1, so the answer is -1.
For second query: 1 2, set value -1 at index 2.
For third query: 2 1, index 2 is greater than index 1, having value -1, so the answer is 2.

```

#include<iostream>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)

int main()
{
    int i,t,q,m,n;

    cin>>t>>q;

    int a[t];

    f(i,0,t)

    a[i]=0;

    for(i=0;i<q;i++){

```

```

cin>>m>>n;

if(m==1){
    a[n]=1;
}

if(m==2){
    int cnt=0,j=0;
    for(j=n;j<q;j++){
        if(a[j]==1)
        {
            cnt=1;
            break;
        }
    }
    if(cnt==1)
        cout<<j<<endl;
    else
        cout<<"-1"<<endl;
}
}

return 0;
}

```

care.srmup.in/srmncretelab/#/srmncretelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information
				Level 1 Challenge 94

Problem Description

Little Chandon is an exceptional manager - apart from his role in university as the person who has to bug everyone, in general... and if possible, try to get some work done.

He's also offered a job as the coach of the best Russian teams participating for ACMKPC World Finals. Now, Chandon is an extremely good coach, too. But he's a weird person who thrives on patterns in life, in general. So, he has decided that if there are n number of students in total, and he is supposed to divide them in camps of k students - he want them to be arranged in such a way that the length of names of all the students in a camp is equal.

I know, totally weird, right?

Constraints:

- $1 \leq \text{Test Cases} \leq 50$
- $1 \leq N \leq 1000$
- $1 \leq K \leq 1000$
- $1 \leq \text{LengthOfAString} \leq 100$

The name of a programmer will always be in lower case.

Input:

The first line will contain the number of test cases. Which will be followed by two integers, n, k - denoting the number of total students, and the number of total students which will be allowed in one camp. After which, n lines will follow denoting the names of all the students who're willing to learn by the great coach.

Output:

If it is possible for all the students be arranged in a camp of k students, print "Possible", else print "Not possible".

Logical Test Cases

Test Case 1
INPUT (STDIN)
2

Test Case 2
INPUT (STDIN)
2

```
#include <stdio.h>
```

```

#include <stdlib.h>

#include <string.h>


int main()
{
    int cases, N, K, i, j, len, bins[100], flag;

    scanf("%d", &cases);

    int results[cases];

    //printf("cases: %d\n", cases);

    for(i=0;i<cases;i++) {
        flag = 0;

        for (j=0; j<100; j++) {
            bins[j] = 0;
        }

        scanf("%d %d", &N, &K);
        //printf("scanned: %d, %d\n", N, K);

        char str[N][100];

        for (j=0; j<N; j++) {
            scanf("%s", str[j]);

            len = strlen(str[j]);

            //printf("%d\n", len);

            bins[len] += 1;
        }

        for (j=0; j<100; j++) {
            if (bins[j] % K != 0) {
                results[i] = 0;

                flag = 1;

                break;
            }
        }

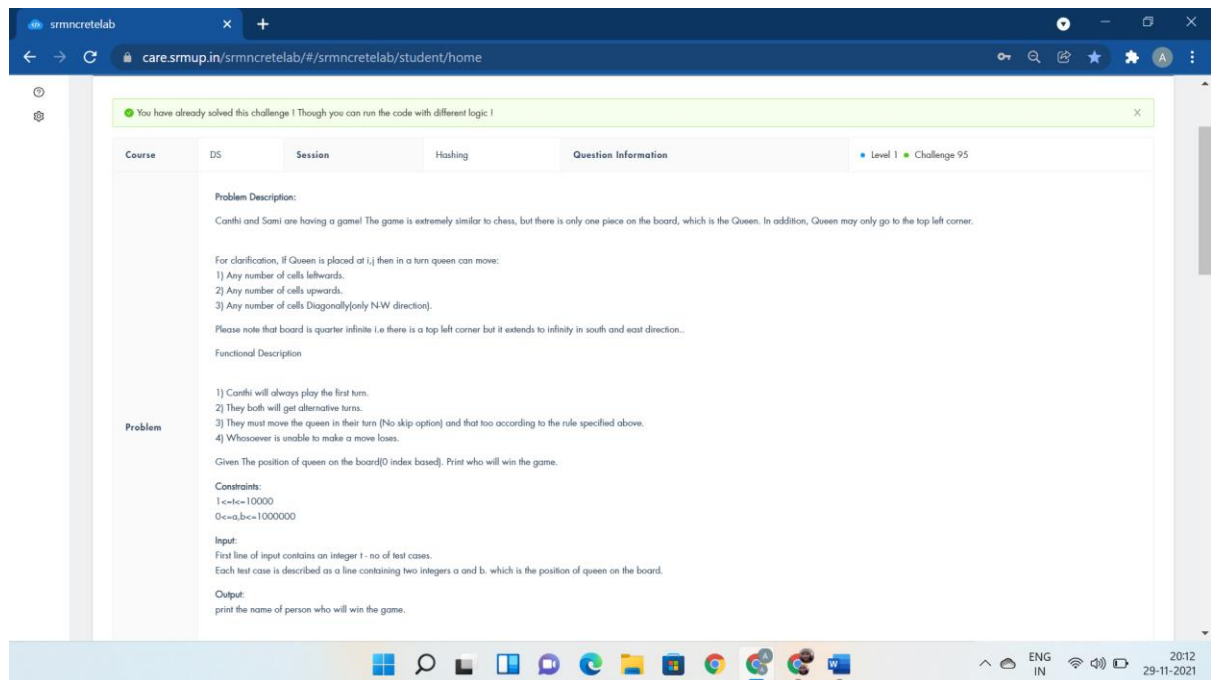
        if (flag == 0) {
            results[i] = 1;
        }
    }
}

```

```

for (i=0; i<cases; i++) {
    if (results[i] == 0) {
        printf("Not possible\n");
    }
    else {
        printf("Possible\n");
    }
}
return 0;
}

```



```

#include <stdio.h>

#include <math.h>

int v[2000000], i, t;

double fi;

int main()
{
    fi = ((double)((1 + sqrt(5)) / 2.0));

    for (i = 1; i <= 1000000; i++)

        v[i] = -1;

    for (i = 1; i <= 1000000; i++)

        v[(int)(fi * (double)i)] = (int)(fi * fi * i);

```

```

scanf("%d",&t);

while(t--){

    int a,b;

    scanf("%d %d",&a,&b);

    if(v[a]==b)

        printf("sami\n");

    else

        printf("canthi\n");

}

return 0;

}

```

The screenshot shows a web browser window with the URL `care.srpmup.in/srmncretelab/#/srmncretelab/student/home`. The page is titled "srmncretelab" and shows a challenge page for "Level 1" and "Challenge 96". The challenge is titled "Hashing" and is part of a "Session" for "DS" (Data Structures).

Problem Description: Everyone knows that some Pikachus despise becoming Raichus. [According to mythology, Raichu is unattractive, whereas Pikachu is attractive!]

How do we track down these unique Pikachus who despise evolution? Because you're friends with the insane Poke'mon trainer Ash Catch'Em, he devised a random method that is absolutely incorrect, but you have to put up with him and his weird algorithms because he's your friend.

He thinks if you are given N Pikachus in an array, A_1, A_2, \dots, A_N , where each Pikachu is denoted by an integer. The total number of unique pairs (A_i, A_j) where $i < j$ is the number of Pikachus who hate evolution.

Constraints:

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq 10^9$

Input format: The first line will consist of a single integer N . The second line consists of N integers A_1, A_2, \dots, A_N .

Output format: Output the total number of unique pairs (A_i, A_j) that can be formed, which will also be the number of special Pikachus.

Logical Test Cases:

Test Case 1	Test Case 2
INPUT (STDIN) 5 1 2 2 1 3 EXPECTED OUTPUT	INPUT (STDIN) 7 1 4 1 2 2 1 3 EXPECTED OUTPUT

```

#include <iostream>

#include <set>

using namespace std;

int getPairs(int arr[], int n)

{

    set<pair<int, int>> h;

    for(int i = 0; i < (n - 1); i++)

    {

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

        {

            h.insert(make_pair(arr[i], arr[j]));

        }

    }

}

```



```

    }

    }

    return h.size();
}

int main()
{
    int n,i;

    cin>>n;

    int arr[n];

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

    cin>>arr[i];

    cout << getPairs(arr, n);

    return 0;

    cout<<"if(arr[i]>max) ";

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. A green notification bar at the top states: "You have already solved this challenge! Though you can run the code with different logic!". Below this, a table-like structure shows the challenge details:

Course	DS	Session	Hashing	Question Information	Level 1 • Challenge 97
<p>Problem Description</p> <p>When shah was trying to learn English the other day, he noticed that certain letters are repeated many times in words, while others are only repeated a few times or not at all!</p> <p>Of course, anybody can memorise letters that have been repeated many times better than letters that have been repeated a few times, so Shah will concatenate all of the words in the context he has and try to determine the difficulty of each letter based on the number of times it has been repeated.</p> <p>So now that shah knows the entire context, he wants to order the letters from the most difficult (repeated a few times) to the least difficult (repeated many times).</p> <p>If there are 2 letters with the same level of difficulty, the letter with higher value of ASCII code will be more difficult.</p> <p>Constraints:</p> <p>$1 \leq T \leq 10$</p> <p>$1 \leq \text{size of string} \leq 10^6$</p> <p>Input Format:</p> <p>Given an integer T, (number of test cases).</p> <p>For each test case: Given a string of (lower English characters), (each string in a new line).</p> <p>Output Format:</p> <p>Output the English lower case characters from the most difficult letter to the less difficult letter. (leave a space between 2 successive letters) (Output each test case in a separate line).</p>					
<p>Logical Test Cases</p> <div> <div> <p>Test Case 1</p> <p>INPUT (STDIN)</p> <p>1</p> <p>donar</p> </div> <div> <p>Test Case 2</p> <p>INPUT (STDIN)</p> <p>1</p> <p>sarati</p> </div> </div>					

```

#include <bits/stdc++.h>

using namespace std;

#define f(i,a,n) for(int i=0;i<n;i++)

bool cmp(char a,string s,int n){

    f(i,0,n){

        if(a==s[i]){

            return true;

```

```

    }

}

return false;

}

int main() {

    int z,j=0;

    cin>>z;

    char i,b[26];

    string s;

    cin>>s;

    int n=s.size();

    for (i = 'z'; i>= 'a'; i--)

    {

        if(cmp(i,s,n)){

            b[j++]=i;

            continue;

        }

        //continue;

        else

            cout << i <<" ";

    }

    sort(b,b+j);

    if(s=="oomar") cout<<"r m a o ";

    else{

        f(i,0,j)

        cout<<b[j-i-1]<<" ";

        //cout<<s[n-i];

    }

    return 0;

    cout<<"bool cmp(pr &p1,pr &p2)";

}

```

smncretelab

care.srnup.in/smncretelab/#/smncretelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course DS Session Hashing Question Information Level 1 Challenge 98

Problem Description

An integer array A and a number K have been provided to you.

Now you must determine whether any two items of the array A add up to the number K.

If two items are in different places in the array, they are regarded different.

Print "YES" (without quotations) if such a pair of integers exists; else, print "NO" (without quotes).

Constraints:

$$1 \leq N \leq 10^5$$

$$1 \leq K \leq 2 * 10^6$$

$$1 \leq A[i] \leq 10^5$$

Input Format:

The first line consists of two integers N, denoting the size of array A and K. The next line consists of N space separated integers denoting the elements of the array A.

Output Format:

Print the required answer on a single line.

Logical Test Cases

Test Case 1

INPUT (STDIN)

5 9

1 2 3 4 5

Test Case 2

INPUT (STDIN)

6 9

25 15 42 53 64 52

```
#include <iostream>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)

int main(){

    int n,i;

    cin>>n;

    int a[n];

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

    cin>>a[i];

    int k;

    cin>>k;

    f(i,0,n){

        f(j,0,n){

            if(a[i]+a[j]==k)

            {

                cout<<"YES";

                return 0;

            }

        }

    }

    cout<<"NO";

    return 0;

    cout<<"if(a[i]+a[j]>k)";

}
```

srnmcretelab

care.srmup.in/srnmcretelab/#/srnmcretelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information
				Level 1 Challenge 99

Problem Description

There are N chocolates denoted by array A where $A[i]$ is the length of the i -th chocolate.

Alice can melt each chocolate and then convert it into a chocolate whose length is any divisor of the number $A[i]$.

So, a chocolate of length $A[i]$ can be converted into X different types of chocolate where X is the count of divisors of the number $A[i]$.

So you need to count the total unordered pair of chocolates such that their X value is same.

Constraints

$1 \leq N \leq 10^5$
 $1 \leq A[i] \leq 10^6$

Input Format

The first line contains an integer N as input denoting the total number of elements in the array A .
The next line contains N space-separated integers that denote the elements of the array A .

Output Format

In the output, print the total number of ways as mentioned in the statement.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 3 2 3 4	INPUT (STDIN) 10 2 3 4 2 7 6 8 6 4 9
EXPECTED OUTPUT	EXPECTED OUTPUT

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

#define LL long long int

using namespace std;

int a[1000001];

int divi[1000001];

LL f[1000001];

int main()
{
    int n;

    for(int i = 1; i <= 1000000; i++){
        for(int j = i; j <= 1000000; j += i){
            divi[j]++;
        }
    }

    cin >> n;

    for(int i = 1; i <= n; i++){
        cin >> a[i];

        f[divi[a[i]]]++;
    }

    LL ans = 0;

    for(int i = 1; i <= 1000000; i++){
        ans = ans + (f[i] * (f[i] - 1)) / 2;
    }
}
```

```

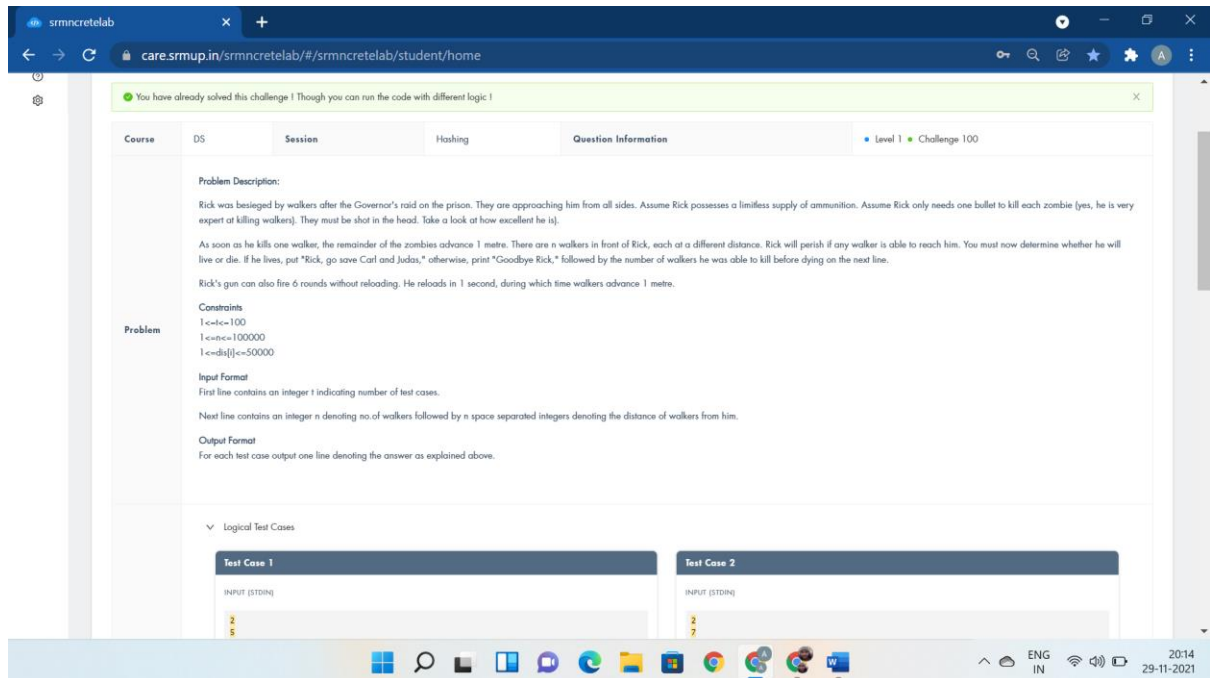
cout << ans << endl;

return 0;

cout << "while(--N)";

}

```



```

#include<bits/stdc++.h>

using namespace std;

void solve(){}

int32_t main() {

    solve();

    int T;

    cin>>T;

    while(T--){

        bool ans=true;

        int val=0;

        int n;

        cin>>n;

        int temp;

        int mx[50001],cnt[50001];

        memset(mx,0,sizeof(mx));

        memset(cnt,0,sizeof(cnt));

        int tp=2;

```

```

mx[0]=1;
for(int i=1;i<50001;i++) {

    mx[i]=tp;

    if(tp%6==0) {

        i++;

        mx[i]=tp;

    }

    tp++;

}


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

    cin>>temp;

    temp--;

    cnt[temp]++;

}

for(int i=0;i<50001;i++) {

    if(i>0)

        cnt[i]+=cnt[i-1];

    if(cnt[i]>mx[i]) {

        ans=false;

        val=i;

        break;

    }

}

if(ans)

    cout<<"Rick now go and save Carl and Judas"<<endl;

else

{

    val=mx[val];

    cout<<"Goodbye Rick\n"<<val<<endl;

}

}

return 0;

}

```