TIẾN HÓA POKEMON

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07020/top)

Rảnh rỗi, Tí đang ngồi chơi Pokemon Go, một game đã không còn mấy thịnh hành. Vì vậy, bắt pokemon dễ như ăn kẹo. Tí bắt được rất nhiều con pokemon và đầu tư mua được một số ít viên đá tiến hóa. Tuy nhiên, mỗi loại đá chỉ dùng được cho một loại pokemon. Với một con pokemon chủng P[i], Tí cần dùng K[i] viên đá để tiến hóa cho nó. Sau khi tiến hóa xong, Tí sẽ thu lại được thêm 2 viên đá nữa.

Không có việc gì làm nên Tí cho đống pokemon của mình tiến hóa hết một lượt. Các bạn hãy tính xem Tí có thể tiến hóa được bao nhiêu con, và loại pokemon nào được tiến hóa nhiều nhất?

**Input**

Dòng đầu tiên là N (N ≤ ≤ 70), số lượng loại pokemon mà Tí bắt được. Giả sử rằng Tí có vô hạn các con pokemon trong mỗi loại.

2N dòng tiếp theo, mỗi dòng gồm một xâu P[i], là loại pokemon mà Tí có. Tiếp theo là 2 số nguyên K[i] (10 ≤ K[i] ≤ 400) và M[i] (1 ≤ M[i] ≤ 10 000), lần lượt là số lượng viên đá cần để tiến hóa cho một con pokemon loại P[i], và số lượng viên đá loại i mà Tí có.

**Output**

Dòng đầu tiên in ra tổng số lượng các con pokemon được tiến hóa.

Dòng thứ hai in ra loại pokemon nào được tiến hóa nhiều nhất. Nếu có nhiều đáp án, hãy in ra tên loại pokemon xuất hiện sớm hơn trong danh sách.

**Test ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 5  Pidgey  12 36  Ivysaur  10 63  Charmander  35 96  Pikachu  12 32  Squirtle  15 10 | 15  Ivysaur |

#include<stdio.h>

#include<string.h>

int b[10000]={0};

int main(){

int n,m,k;

char a[1001][1001];

scanf("%d",&n);

int i,max=0,s=0;

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

scanf("\n");

gets(a[i]);

scanf("%d %d",&k,&m);

int e=0;

while(m>=k){

int d=m/k;

e=e+d;

m=m%k+2\*d;

}

b[i]=e;

if(b[i]>max) max=b[i];

s=s+b[i];

}

printf("%d\n",s);

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

if(b[i]==max){

puts(a[i]);

break;

}

}

}

ƯỚC SỐ NGUYÊN TỐ LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03045/top)

Cho số nguyên dương N. Hãy đưa ra ước số nguyên tố lớn nhất của N.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test ghi số nguyên dương N.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 2≤N≤1010.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 2  315  31 | 7  31 |

#include<stdio.h>

#include<math.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

long long n,x,i;

scanf("%lld",&n);

for(i=2;i<=sqrt(n);i++){

while(n%i==0){

x=i;

n=n/i;

}

}

if(n>1) x=n;

printf("%lld\n",x);

}

}

TÍNH TỔNG NHỎ NHẤT VÀ LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03057/top)

Cho hai số nguyên dương X1, X2. Ta chỉ được phép thay đổi chữ số 5 thành 6 và ngược lại chữ số 6 thành chữ số 5 của các số X1 và X2. Hãy đưa ra tổng nhỏ nhất và tổng lớn nhất các số X1 và X2 được tạo ra theo nguyên tắc kể trên.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là cặp các số X1, X2.
* T, X1, X2 thỏa mãn ràng buộc: 1≤ T ≤100; 0≤ X1, X2 ≤1018.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 645  666 5466 4555 | 1100  1312 10010 11132 |
|  |  |

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

long long a,b;

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

long long s=a+b,x=1,y=1,s1=0,s2=0;

int d,e;

while(a>0){

d=a%10;

if(d==6) {s1=s1+x;}

if(d==5) {s2=s2+x;}

a=a/10;

x=x\*10;

}

while(b>0){

e=b%10;

if(e==6) {s1=s1+y;}

if(e==5) {s2=s2+y;}

b=b/10;

y=y\*10;

}

long long sl=s+s2,sn=s-s1;

printf("%lld %lld\n",sn,sl);

}

}

QUẢN LÝ SINH VIÊN - 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07021/top)

Thông tin về sinh viên gồm có:

1. mã sinh viên là số nguyên tự động tăng;

2. tên sinh viên;

3. điểm sinh viên là số thực gồm điểm môn A, môn B, môn C.

Viết chương trình thực hiện 3 chức năng:

1. thêm mới

2. cập nhật điểm theo mã sinh viên

3. hiển thị danh sách sinh viên theo tổng điểm tăng dần

**Input:**

- Dòng thứ nhất là lựa chọn chức năng thực hiện

- Dòng thứ hai

+ Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sinh viên sẽ thêm mới

+ Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sinh viên sẽ sửa đổi

- Dòng tiếp theo là thông tin sinh viên

Chức năng 3 cũng là kết thúc của luồng input

**Output:**

- Nếu lựa chọn là 1 thì in ra số sinh viên được thêm

- Nếu lựa chọn là 2 thì in ra mã sinh viên được sửa

- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2  nguyen van hai  8.5 5.5 7.5  tran van tuan  8.5 .50 9.0  2  2  tran van nam  5.5 5.0 6.0  3 | 2  2  2 tran van nam 5.5 5.0 6.0  1 nguyen van hai 8.5 5.5 7.5 |

#include<stdio.h>

#include<string.h>

struct sv{

int stt;

char ten[1001];

float d1,d2,d3;

float tong;

};

int main(){

int p=1,b[1001];

int t,kt=1;

struct sv a[1001],x;

while(kt!=0){

scanf("%d",&b[p]);

if(b[p]==1){

scanf("%d",&t);

for(int i=1;i<=t;i++){

scanf("\n");

gets(a[i].ten);

scanf("\n");

scanf("%f %f %f",&a[i].d1,&a[i].d2,&a[i].d3);

a[i].stt=i;

a[i].tong=a[i].d1+a[i].d2+a[i].d3;

}

printf("%d\n",t);

}

else if(b[p]==2){

scanf("%d",&x.stt);

scanf("\n");

gets(x.ten);

scanf("\n");

scanf("%f %f %f",&x.d1,&x.d2,&x.d3);

x.tong=x.d1+x.d2+x.d3;

printf("%d\n",x.stt);

for(int i=1;i<=t;i++){

if(a[i].stt==x.stt ) a[i]=x;

}

for(int i=1;i<t;i++){

for(int j=i+1;j<=t;j++){

if(a[j].tong<a[i].tong){

struct sv tg=a[j];

a[j]=a[i];

a[i]=tg;

}

}

}

}

else break;

p++;

}

for(int i=1;i<=t;i++){

printf("%d %s %.1f %.1f %.1f\n",a[i].stt,a[i].ten,a[i].d1,a[i].d2,a[i].d3);

}

}

QUẢN LÝ SINH VIÊN - 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07022/top)

Thông tin về sinh viên gồm có:

1. mã sinh viên là số nguyên tự động tăng;

2. tên sinh viên;

3. điểm sinh viên là số thực gồm điểm môn A, môn B, môn C.

Viết chương trình thực hiện 3 chức năng:

1. thêm mới

2. cập nhật điểm theo mã sinh viên

3. hiển thị danh sách các sinh viên có điểm 3 môn tăng dần (theo thứ tự xuất hiện)

**Input:**

- Dòng thứ nhất là lựa chọn chức năng thực hiện

- Dòng thứ hai

+ Nếu lựa chọn ở dòng thứ nhất là 1 thì nhập số sinh viên sẽ thêm mới

+ Nếu lựa chọn ở dòng thứ nhất là 2 thì nhập mã sinh viên sẽ sửa đổi

- Dòng tiếp theo là thông tin sinh viên

Chức năng 3 cũng là kết thúc của luồng input

**Output:**

- Nếu lựa chọn là 1 thì in ra số sinh viên được thêm

- Nếu lựa chọn là 2 thì in ra mã sinh viên được sửa

- Nếu lựa chọn là 3 thì in ra danh sách theo mô tả dưới đây (các thông tin cách nhau một khoảng trắng)

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2  nguyen van hai  8.5 5.5 7.5  tran van tuan  8.5 6.5 9.0  2  2  tran van nam  5.5 6.0 7.0  3 | 2  2  2 tran van nam 5.5 6.0 7.0 |

#include<stdio.h>

#include<string.h>

struct sv{

int stt;

char ten[1001];

float d1,d2,d3;

float tong;

};

int main(){

int p=1,b[1001];

int t,kt=1;

struct sv a[1001],x;

while(kt!=0){

scanf("%d",&b[p]);

if(b[p]==1){

scanf("%d",&t);

for(int i=1;i<=t;i++){

scanf("\n");

gets(a[i].ten);

scanf("\n");

scanf("%f %f %f",&a[i].d1,&a[i].d2,&a[i].d3);

a[i].stt=i;

a[i].tong=a[i].d1+a[i].d2+a[i].d3;

}

printf("%d\n",t);

}

else if(b[p]==2){

scanf("%d",&x.stt);

scanf("\n");

gets(x.ten);

scanf("\n");

scanf("%f %f %f",&x.d1,&x.d2,&x.d3);

x.tong=x.d1+x.d2+x.d3;

printf("%d\n",x.stt);

for(int i=1;i<=t;i++){

if(a[i].stt==x.stt ) a[i]=x;

}

for(int i=1;i<t;i++){

for(int j=i+1;j<=t;j++){

if(a[j].tong<a[i].tong){

struct sv tg=a[j];

a[j]=a[i];

a[i]=tg;

}

}

}

}

else break;

p++;

}

for(int i=1;i<=t;i++){

if(a[i].d1<a[i].d2 && a[i].d2<a[i].d3){

printf("%d %s %.1f %.1f %.1f\n",a[i].stt,a[i].ten,a[i].d1,a[i].d2,a[i].d3);

}

}

}

CHIA HẾT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03060/top)

Cho hai số nguyên dương n và k. Hãy kiểm tra xem giai thừa của n (n!) có chia hết cho 2k hay không.

**Input**

Có một dòng ghi 2 số n và k (1 ≤ n, k ≤ 100).

**Output**

Nếu n! chia hết cho 2k thì in ra “Yes”, ngược lại in ra “No”.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3 | Yes |
| **Input** | **Output** |
| 1 1 | No |

#include<stdio.h>

int main(){

int n,k,dem=0;

scanf("%d %d",&n,&k);

for(int i=2;i<=n;i++){

int x=i;

while(x%2==0){

dem++;

x/=2;

}

}

if(dem>=k)printf("Yes");

else printf("No");

}

SỐ THUẦN NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03025/top)

Một số được coi là thuần nguyên tố nếu nó là số nguyên tố, tất cả các chữ số là nguyên tố và tổng chữ số của nó cũng là một số nguyên tố.  Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số thuần nguyên tố.

**Input:**Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số.

**Output:**Mỗi bộ test viết ra số lượng các số thuần nguyên tố tương ứng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 2  23 199  2345 6789 | 1  15 |

#include<stdio.h>

#include<math.h>

int snt(long a){

int kt=0;

if(a<2) return 0;

for(long i=2;i<=sqrt(a);i++){

if(a%i==0){

kt=1;

break;

}

}

if(kt==1 ) return 0;

else return 1;

}

int ktcsnt(long a){

int d,kt=0;

while(a>0){

d=a%10;

a=a/10;

if(d!=2 && d!=3 && d!=5 && d!=7){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int kttongsnt(long a){

long long s=0,d;

while(a>0){

d=a%10;

s=s+d;

a=a/10;

}

if(snt(s)==0) return 0;

else return 1;

}

int main(){

int t,k;

scanf("%d",&t);

for(k=0;k<t;k++){

long n,m,i,d=0;

scanf("%ld %ld",&n,&m);

for(i=n;i<=m;i++){

if( ktcsnt(i)==1 ){

if( kttongsnt(i)==1){

if(snt(i)==1){

d++;

}

}

}

}

printf("%ld\n",d);

}

return 0;

}

GHÉP HÌNH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01070/top)

Cho ba hình chữ nhật. Các bạn được phép xoay hình nhưng không được phép xếp chồng lấn lên nhau, hỏi 3 hình chữ nhật đó có thể ghép thành một hình vuông được hay không

**Input:** Có ba dòng, mỗi dòng ghi hai số nguyên dương là chiều rộng và chiều cao của hình chữ nhật (các số đều không quá 100).

**Output:** Ghi ra YES nếu có thể tạo thành hình vuông, NO nếu không thể.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 2  1 6  7 6 | YES |

#include<stdio.h>

int main(){

int a,b,m,n,x,y;

scanf("%d%d%d%d%d%d",&a,&b,&m,&n,&x,&y);

if(m+x==a && b+n==a && y+b==a) printf("YES");

else if(a+x==m && b+n==m && y+n==m) printf("YES");

else if(a+m==x && b+y==x && n+y==x) printf("YES");

else if(b==n && n==y && a+m+x==b) printf("YES");

else if(a+x==y && m+x==y && b+n==y ) printf("YES");

else if(x+a==b && m+a==b && y+n==b) printf("YES");

else if(x+m==n && a+m==n && y+b==n) printf("YES");

else if(a==x && m==x && b+y+n==a) printf("YES");

else if(m+x==b && a+n==b && y+a==b) printf("YES");

else if(b+x==m && a+n==m && y+n==m) printf("YES");

else if(b+m==x && a+y==x && n+y==x) printf("YES");

else if(a==n && n==y && b+m+x==a) printf("YES");

else if(b+x==y && m+x==y && a+n==y ) printf("YES");

else if(x+b==a && m+b==a && y+n==a) printf("YES");

else if(x+m==n && b+m==n && y+a==n) printf("YES");

else if(b==x && m==x && a+y+n==b) printf("YES");

else if(n+x==a && b+m==a && y+b==a) printf("YES");

else if(a+x==n && b+m==n && y+m==n) printf("YES");

else if(a+n==x && b+y==x && m+y==x) printf("YES");

else if(b==m && m==y && a+n+x==b) printf("YES");

else if(a+x==y && n+x==y && b+m==y ) printf("YES");

else if(x+a==b && n+a==b && y+m==b) printf("YES");

else if(x+n==m && a+n==m && y+b==m) printf("YES");

else if(a==x && n==x && b+y+m==a) printf("YES");

else if(m+y==a && b+n==a && x+b==a) printf("YES");

else if(a+y==m && b+n==m && x+n==m) printf("YES");

else if(a+m==y && b+x==y && n+x==y) printf("YES");

else if(b==n && n==x && a+m+y==b) printf("YES");

else if(a+y==x && m+y==x && b+n==x ) printf("YES");

else if(y+a==b && m+a==b && x+n==b) printf("YES");

else if(y+m==n && a+m==n && x+b==n) printf("YES");

else if(a==y && m==y && b+x+n==a) printf("YES");

else printf("NO");

}

LIỆT KÊ VÀ ĐẾM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04032/top)

Cho một dãy các số nguyên dương không quá 9 chữ số, mỗi số cách nhau vài khoảng trống, có thể xuống dòng. Hãy tìm các số không giảm (các chữ số theo thứ tự từ trái qua phải tạo thành dãy không giảm) và đếm số lần xuất hiện của các số đó.

**Input:** Gồm các số nguyên dương không quá 9 chữ số. Không quá 100000 số.

**Kết quả** Ghi ra các số không giảm kèm theo số lần xuất hiện. Các số được liệt kê theo thứ tự sắp xếp số lần xuất hiện giảm dần.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 123 321 23456 123 123 23456 3523 123 321 8988 7654 9899 3456 123 999 3456 987654321 4546 63543 4656 13432 4563 123471 659837 454945 34355 9087 9977 98534 3456 23134 | 123 5  3456 3 23456 2 999 1 |

#include<stdio.h>

int thuannghich(int n){

int a,b;

while(n>=10){

a=n%10; b=(n/10)%10;

if(a<b) return 0;

n=n/10;

}

return 1;

}

int tim(int a[],int n, int x){

for(int i=0;i<n;i++) if(a[i]==x) return i;

return -1;

}

void sapxep(int a[],int b[],int n){

int i,j,t;

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

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

if(b[i]<b[j]){

t=a[i]; a[i]=a[j]; a[j]=t;

t=b[i]; b[i]=b[j]; b[j]=t;

}

}

void in(int a[],int b[],int n){

for(int i=0;i<n;i++) printf("%d %d\n",a[i],b[i]);

}

main(){

int a[10000],b[10000]={0},x,n=0;

while(scanf("%d",&x)!=-1){

if(thuannghich(x)){

int t = tim(a,n,x);

if(t==-1){

a[n] = x; b[n] = 1; n++;

}

else b[t]++;

}

}

sapxep(a,b,n); in(a,b,n);

}

SẮP XẾP NỔI BỌT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04029/top)

Hãy thực hiện thuật toán sắp xếp nổi bọt trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Input:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Output:**Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  5 3 2 7 | Buoc 1: 3 2 5 7  Buoc 2: 2 3 5 7 |

#include<stdio.h>

int main (){

int n;

scanf("%d",&n);

int a[n],i,j;

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

scanf("%d",&a[i]);

}

int k=1;

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

int kt=0;

int x=0;

for(j=x+1;j<n;j++){

if(a[x]>a[j]){

int d=a[j];

a[j]=a[x];

a[x]=d;

kt++;

}

x++;

}

if(kt==0) {

break;

}

printf("Buoc %d: ",k);

for(j=0;j<n;j++){

printf("%d ",a[j]);

}

printf("\n");

k++;

}

}

LEO NÚI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04035/top)

Có N (1≤N≤25000) người leo lên và leo xuống trên 1 ngọn núi. Người i mất U(i) thời gian leo lên và D(i) thời gian để leo xuống. Trong một thời điểm chỉ có tối đa người 1 người có thể lên và tối đa 1 người có thể xuống (có thể 1 người lên, 1 người xuống). Những người khác có thể đứng chờ ở đỉnh ngọn núi. Thứ tự đi xuống có thể khác thứ tự đi lên. Bạn hãy xác định xem thời gian tối thiểu để cho N người lên và xuống ngọn núi là bao nhiêu.

**Input:**Dòng 1 ghi số N. N dòng tiếp theo chứa 2 số U(i) và D(i) (1 ≤ U(i) , D(i) ≤ 50000)

**Output:**Ghi ra thời gian tối thiểu có thể.

**Ví dụ:***(Giải thích: đi lên và xuống theo thứ tự người 3->1->2)*

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  6 4  8 1  2 3 | 17 |

#include<stdio.h>

int main()

{

long n;

long MinU=50005;

long MinD=50005;

long U,D;

long TU=0,TD=0;

scanf("%ld",&n);

for(long i=1;i<=n;i++) { scanf("%ld%ld",&U,&D); TU+=U; TD+=D; if(MinU>U) MinU=U;

if(MinD>D) MinD=D;

}

TU+=MinD;

TD+=MinU;

if(TU>TD) printf("%ld",TU);

else printf("%ld",TD);

}

 DÃY CON LIÊN TIẾP CÓ TỔNG LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04040/top)

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy tìm một dãy con liên tiếp sao cho tổng các phần tử của chúng là lớn nhất.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 10). Mỗi test gồm số nguyên N (1≤ N ≤ 100 000), số lượng phần tử trong dãy số ban đầu. Dòng tiếp theo gồm N số nguyên A[i] (-109 ≤ A[i] ≤ 109).

**Output:**Với mỗi test, in ra một số nguyên là đáp án của bài toán trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  8  -2 -3 4 -1 -2 1 5 -3  5  1 2 3 4 5 | 7  15 |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

for(k=0;k<t;k++){

long n,j,i,max,kt=0;

scanf("%ld",&n);

long a[n];

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

scanf("%ld",&a[i]);

max=a[0];

if(max<a[i]) max=a[i];

}

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

if(a[i]>0){

kt=1;

break;

}

}

if(kt==0) printf("%ld\n",max);

else{

long s=0,smax=a[0];

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

if((s+a[i])<0){

s=0;

continue;

}

s+=a[i];

if(s>smax) smax=s;

}

printf("%ld\n",smax);

}

}

}

SỐ ĐẦU TIÊN BỊ LẶP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04042/top)

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy tìm số xuất hiện nhiều hơn 1 lần trong dãy số và số thứ tự là nhỏ nhất.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 10). Mỗi test gồm số nguyên N (1≤ N ≤ 100 000), số lượng phần tử trong dãy số ban đầu. Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 109).

**Output:**Với mỗi test in ra đáp án của bài toán trên một dòng. Nếu không tìm được đáp án, in ra “NO”.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  7  10 5 3 4 3 5 6  4  1 2 3 4 | 5  NO |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

for(k=0;k<t;k++){

int n,j,i,kt=0,d;

scanf("%d",&n);

long a[n];

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

scanf("%ld",&a[i]);

}

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

d=0;

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

if(a[j]==a[i]){

d++;

break;

}

}

if(d>0){

printf("%ld",a[i]);

kt=1;

break;

}

}

if(kt==0) printf("NO");

printf("\n");

}

}

BRT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04046/top)

Thành phố X có N thị trấn trên trục đường chính. Tọa độ của các thị trấn lần lượt là a[1], a[2], …, a[N], các tọa độ này là phân biệt, không có 2 tọa độ nào trùng nhau.

Chính quyền thành phố muốn xây dựng một tuyến buýt nhanh BRT để kết nối 2 thị trấn gần nhau nhất với nhau.

Bạn hãy tính thử xem chiều dài của tuyến buýt này bằng bao nhiêu? Và có bao nhiêu cặp thị trấn có tiềm năng giống nhau để xây dựng tuyến BRT này.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test bắt đầu bằng số nguyên N (N ≤ 100 000).

Dòng tiếp theo gồm N số nguyên A[i] (-109 ≤ A[i] ≤ 109).

**Output:**

Với mỗi test in ra 2 số nguyên C và D, lần lượt là khoảng cách ngắn nhất giữa 2 thị trấn, và số lượng cặp thị trấn có cùng khoảng cách ngắn nhất này.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  6 -3 0 4  3  -2 0 2 | 2 1  2 2 |

**#include <stdio.h>**

**#include <stdlib.h>**

**void merge(long arr[], long l, long m, long r)**

**{**

**long i, j, k;**

**long n1 = m - l + 1;**

**long n2 = r - m;**

**long L[n1], R[n2];**

**for (i = 0; i < n1; i++)**

**L[i] = arr[l + i];**

**for (j = 0; j < n2; j++)**

**R[j] = arr[m + 1 + j];**

**i = 0;**

**j = 0;**

**k = l;**

**while (i < n1 && j < n2) {**

**if (L[i] <= R[j]) {**

**arr[k] = L[i];**

**i++;**

**}**

**else {**

**arr[k] = R[j];**

**j++;**

**}**

**k++;**

**}**

**while (i < n1) {**

**arr[k] = L[i];**

**i++;**

**k++;**

**}**

**while (j < n2) {**

**arr[k] = R[j];**

**j++;**

**k++;**

**}**

**}**

**void mergeSort(long arr[], long l, long r)**

**{**

**if (l < r) {**

**long m = l + (r - l) / 2;**

**mergeSort(arr, l, m);**

**mergeSort(arr, m + 1, r);**

**merge(arr, l, m, r);**

**}**

**}**

**int main(){**

**int t;**

**scanf("%d",&t);**

**while(t--){**

**long n,i,j;**

**scanf("%ld",&n);**

**long arr[n+1];**

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

**scanf("%ld",&arr[i]);**

**}**

**long long min=9999999999,dem=0;**

**mergeSort(arr, 0, n-1);**

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

**if(arr[i+1]-arr[i]<min) min=arr[i+1]-arr[i];**

**}**

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

**if(arr[i+1]-arr[i]==min) dem++;**

**}**

**printf("%lld %lld\n",min,dem);**

**}**

**}**

CHU VI TAM GIÁC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07005/top)

Cho 3 điểm A, B, C trong không gian 2 chiều Oxy, hãy tính chu vi tam giác được tạo bởi 3 điểm đó.

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test ghi trên 1 dòng 6 số thực có giá trị tuyệt đối không quá 1000 lần lượt là tọa độ của 3 điểm A, B, C.

**Kết quả**

* Nếu 3 điểm không thể tạo thành tam giác thì in ra INVALID
* Nếu 3 điểm tạo thành 1 tam giác thì in ra chu vi của tam giác đó, làm tròn đến 3 chữ số phần thập phân.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  0 0 0 5 0 199  1 1 1 1 1 1  0 0 0 5 5 0 | INVALID  INVALID  17.071 |

**#include<stdio.h>**

**#include<math.h>**

**float dd(float a, float b,float c, float d){**

**float x=(float)sqrt((a-b)\*(a-b) + (c-d)\*(c-d));**

**return x;**

**}**

**int main(){**

**int t;**

**scanf("%d",&t);**

**while(t--){**

**float xA,yA,xB,yB,xC,yC;**

**scanf("%f %f %f %f %f %f",&xA,&yA,&xB,&yB,&xC,&yC);**

**float a=dd(xA,xB,yA,yB);**

**float b=dd(xB,xC,yB,yC);**

**float c=dd(xC,xA,yC,yA);**

**float m=a+b+c;**

**if((a + b <= c) || (a + c <= b) || (b + c <= a)) printf("INVALID\n");**

**else printf("%.3f\n",m);**

**}**

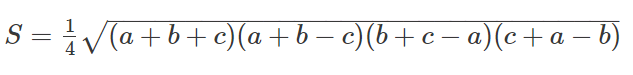
**}**

DIỆN TÍCH TAM GIÁC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07006/top)

Cho 3 điểm A, B, C trong không gian 2 chiều Oxy, hãy tính diện tích tam giác được tạo bởi 3 điểm đó.

*Công thức Heron tính diện tích tam giác khi biết độ dài 3 cạnh là a,b,c:*

****

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test ghi trên 1 dòng 6 số thực có giá trị tuyệt đối không quá 1000 lần lượt là tọa độ của 3 điểm A, B, C.

**Kết quả**

* Nếu 3 điểm không thể tạo thành tam giác thì in ra INVALID
* Nếu 3 điểm tạo thành 1 tam giác thì in ra diện tích của tam giác đó, làm tròn đến 2 chữ số phần thập phân.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  0 0 0 5 0 199  1 1 1 1 1 1  0 0 0 5 5 0 | INVALID  INVALID  12.50 |

**#include <stdio.h>**

**#include <math.h>**

**double canh(double x1, double y1, double x2, double y2) {**

**return sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));**

**}**

**int ktra(double ab, double ac, double bc) {**

**if(ab>=ac+bc || ac>=ab+bc || bc>=ab+ac) return 0;**

**return 1;**

**}**

**double dientich(double ab, double ac, double bc) {**

**return (sqrt((ab+ac+bc)\*(ab+ac-bc)\*(ab-ac+bc)\*(bc-ab+ac)))/4;**

**}**

**int main () {**

**int t;**

**scanf("%d", &t);**

**while(t--) {**

**double x1, y1, x2, y2, x3, y3;**

**scanf("%lf%lf%lf%lf%lf%lf", &x1, &y1, &x2, &y2, &x3, &y3);**

**double ab=canh(x1, y1, x2, y2);**

**double ac=canh(x1, y1, x3, y3);**

**double bc=canh(x2, y2, x3, y3);**

**if(ktra(ab, ac, bc)) printf("%.2lf\n", dientich(ab, ac, bc));**

**else printf("INVALID\n");**

**}**

**return 0;**

**}**

HÌNH VUÔNG NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01025/top)

Cho 2 hình chữ nhật trên mặt phẳng Oxy. Cần tìm hình vuông có kích thước nhỏ nhất sao cho phủ kín được 2 hình chữ nhật đã cho.

**Dữ liệu vào:**

2 dòng, mỗi dòng gồm 4 số nguyên lần lượt mô tả điểm trái dưới và phải trên của hình chữ nhật. Các tọa độ có giá trị tuyệt đối không vượt quá 100.

**Kết quả:**

In ra diện tích của hình vuông tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 6 8 8  1 8 4 9 | 49 |

**#include<stdio.h>**

**int main(){**

**int xA,yA,xB,yB,xC,yC,xD,yD;**

**scanf("%d %d %d %d %d %d %d %d",&xA,&yA,&xB,&yB,&xC,&yC,&xD,&yD);**

**int maxd=xA,mind=xA,maxr=yA,minr=yA;**

**if(xB>maxd) maxd=xB;**

**if(xC>maxd) maxd=xC;**

**if(xD>maxd) maxd=xD;**

**if(xB<mind) mind=xB;**

**if(xC<mind) mind=xC;**

**if(xD<mind) mind=xD;**

**if(yB>maxr) maxr=yB;**

**if(yC>maxr) maxr=yC;**

**if(yD>maxr) maxr=yD;**

**if(yB<minr) minr=yB;**

**if(yC<minr) minr=yC;**

**if(yD<minr) minr=yD;**

**int a=maxd - mind;**

**int b=maxr - minr;**

**if(a>b) printf("%ld",a\*a);**

**else printf("%ld",b\*b);**

**}**

AM GIÁC VUÔNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04043/top)

Theo định lý Pytago, ta đã biết một bộ 3 số (a, b, c) thỏa mãn  a2 + b2 = c2thì đó là ba cạnh của một tam giác vuông.

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là kiểm tra xem ***có tồn tại bộ ba số thỏa mãn là ba cạnh của tam giác vuông hay không.***

**Dữ liệu vào:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test gồm số nguyên N (1≤ N ≤5000).
* Dòng tiếp theo gồm N số nguyên A[i] (1 ≤ A[i] ≤ 109).

**Kết quả:**

Với mỗi test, in ra trên một dòng “YES” nếu tìm được, và “NO” trong trường hợp ngược lại.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5  3 1 4 6 5  3  1 1 1 | YES  NO |

**#include<stdio.h>**

**int kt(long a, long b,long c){**

**if(a\*a==b\*b+c\*c || a\*a+b\*b==c\*c || a\*a+c\*c==b\*b) return 1;**

**else return 0;**

**}**

**int main(){**

**int t;**

**scanf("%d",&t);**

**while(t--){**

**int n,i,j;**

**scanf("%d",&n);**

**long long a[n+1],tg;**

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

**scanf("%lld",&a[i]);**

**a[i]\*=a[i];**

**}**

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

**for(j=i+1;j<n;j++){**

**if(a[i]>a[j]){**

**tg=a[j];**

**a[j]=a[i];**

**a[i]=tg;**

**}**

**}**

**}**

**int x=0,y=n-2,z=n-1,kt=1;**

**while(y>0){**

**if(a[x]+a[y]<a[z]) x++;**

**else if(a[x]+a[y]>a[z]) y--;**

**else {**

**kt=0;**

**break;**

**}**

**if(x==y){**

**z--;**

**y=z-1;**

**x=0;**

**}**

**}**

**if(kt==1) printf("NO\n");**

**else printf("YES\n");**

**}**

**}**

SẮP XẾP MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05025/top)

Cho ma trận A chỉ gồm các số nguyên dương cấp N\*M . Hãy viết chương trình sắp xếp ma trận A theo kiểu đổi chỗ trực tiếp. Một ma trận được coi là sắp xếp nếu góc trên bên trái là nhỏ nhất, góc dưới bên phải là lớn nhất, theo mỗi hàng hay mỗi cột đều được sắp xếp tăng dần.

Hãy thực hiện sắp xếp ma trận theo **ý tưởng đổi chỗ trực tiếp.**

**Dữ liệu vào**

Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: Dòng đầu tiên ghi hai số n và m là bậc của ma trân a; n dòng tiếp theo, mỗi dòng ghi m số của một dòng trong ma trận A.

**Kết quả**

            Với mỗi bộ test ghi ra thứ tự bộ test, sau đó là ma trận tích tương ứng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 3 3 4 2 6 3 5 3 5 7 8 | Test 1:  2 3 4 3 5 6 5 7 8 |

**#include<stdio.h>**

**void nhap(int a[20][20],int n,int m){**

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

**for(int j=0;j<m;j++)**

**scanf("%d",&a[i][j]);**

**}**

**void in(int a[20][20],int n,int m){**

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

**for(int j=0;j<m;j++)**

**printf("%d ",a[i][j]);**

**printf("\n");**

**}**

**}**

**void sapxep(int a[20][20],int n,int m){**

**int i1,i2,j1,j2,t;**

**for(i1=0;i1<n;i1++){**

**for(j1=0;j1<m;j1++){**

**for(i2=i1;i2<n;i2++){**

**for(j2=j1;j2<m;j2++){**

**if(a[i1][j1]>a[i2][j2]){**

**t = a[i1][j1];**

**a[i1][j1] = a[i2][j2];**

**a[i2][j2] = t;**

**}**

**}**

**}**

**}**

**}**

**}**

**void xuly(int a[20][20],int n,int m){**

**sapxep(a,n,m);**

**in(a,n,m);**

**printf("\n");**

**}**

**main(){**

**int a[20][20],n,m,t;**

**scanf("%d",&t);**

**for(int i=1;i<=t;i++){**

**scanf("%d%d",&n,&m);**

**nhap(a,n,m);**

**printf("Test %d:\n",i);**

**xuly(a,n,m);**

**}**

**}**

MA TRẬN XOẮN ỐC 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05017/top)

Cho ma trận A[N][M]. Nhiệm vụ của bạn là in các phần tử của ma trận theo hình xoắn ốc. Ví dụ về in ma trận theo hình xoắn ốc như dưới đây: 1  2  3  4  8  12  16  15  14  13  9  5  6  7  11  10 .

1          2          3          4

5          6          7          8

9          10        11        12

13        14        15        16

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận A[][]; dòng tiếp theo đưa vào N×M số A[i][j].
* T, M, N, A[i][j] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ M, N ≤100; 1≤ A[i][j] ≤105.

**Output:** Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 2 4 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 3 4 1 2 3 4 5 6 7 8 9 10 11 12 | 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 1 2 3 4 8 12 11 10 9 5 6 7 |

**#include<stdio.h>**

**int main(){**

**int t;**

**scanf("%d\n",&t);**

**while(t--){**

**int m,n;**

**scanf("%d %d\n",&m,&n);**

**int a[m][n];**

**int i,j;**

**for(i=0;i<m;i++){**

**for(j=0;j<n;j++){**

**scanf("%d ",&a[i][j]);**

**}**

**}**

**int x=0,y=0,z=0;**

**int p=m\*n;**

**while(-1){**

**for(j=x;j<n;j++){**

**y++;**

**printf("%d ",a[x][j]);**

**}**

**if(y==p) break;**

**for(i=x+1;i<m;i++){**

**y++;**

**printf("%d ",a[i][n-1]);**

**}**

**if(y==p) break;**

**for(j=n-2;j>=x;j--){**

**y++;**

**printf("%d ",a[m-1][j]);**

**}**

**if(y==p) break;**

**for(i=m-2;i>x;i--){**

**y++;**

**printf("%d ",a[i][x]);**

**}**

**if(y==p) break;**

**m--;**

**n--;**

**x++;**

**}**

**printf("\n");**

**}**

**}**

HIỆU HAI SỐ NGUYÊN LỚN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06025/top)

Viết chương trình tính hiệu của hai số nguyên lớn. Chú ý: luôn luôn lấy số lớn hơn trừ đi số nhỏ hơn.

**Input:**

* Dòng 1 ghi số bộ test
* Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một số không quá 1000 chữ số..

**Output:**

      Với mỗi bộ test ghi ra kết quả tính toán tương ứng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 456 789 100000000000000000000000000001 99 123456789012345678901234567890 1234567890 | 333 99999999999999999999999999902 123456789012345678900000000000 |

**#include<stdio.h>**

**#include<string.h>**

**int kt(char a[], char b[]){**

**int la=strlen(a), lb=strlen(b);**

**for(int i=0;i<la;i++){**

**if(a[i]>b[i]) return 1;**

**else if(a[i]<b[i]) return 2;**

**}**

**return 0;**

**}**

**void tru(char a[], char b[]){**

**char c[1000];**

**int l1=strlen(a),l2=strlen(b);**

**for(int i=0;i<l1;i++){**

**a[i]=a[i]-48;**

**}**

**for(int i=0;i<l2;i++){**

**b[i]=b[i]-48;**

**}**

**int e=0,k=0;**

**while(l2>=0){**

**if(a[l1]>=b[l2]){**

**c[k]=a[l1]-b[l2]-e;**

**e=0;**

**}**

**else {**

**c[k]=10+a[l1]-b[l2]-e;**

**e=1;**

**}**

**l1--;**

**l2--;**

**k++;**

**}**

**while(l1>=0){**

**if(a[l1]<e){**

**c[k]=10-e;**

**e=1;**

**}**

**else{**

**c[k]=a[l1]-e;**

**e=0;**

**}**

**k++;**

**l1--;**

**}**

**while(c[k]<1 || c[k]>10){**

**k--;**

**}**

**for(int i=k;i>0;i--){**

**c[i]=c[i]+48;**

**printf("%c",c[i]);**

**}**

**}**

**int main(){**

**int t;**

**char a[1000],b[1000];**

**scanf("%d",&t);**

**while(t--){**

**scanf("\n");**

**gets(a);**

**scanf("\n");**

**gets(b);**

**int n1=strlen(a);**

**int n2=strlen(b);**

**if(n1>n2){**

**tru(a,b);**

**}**

**else if(n2>n1){**

**tru(b,a);**

**}**

**else {**

**if(kt(a,b)==0) printf("0");**

**else if(kt(a,b)==1) {**

**tru(a,b);**

**}**

**else {**

**tru(b,a);**

**}**

**}**

**printf("\n");**

**}**

**}**

ĐỊA CHỈ EMAIL PTIT 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06020/top)

Địa chỉ email của các cán bộ, giảng viên PTIT được tạo ra bằng cách viết đầy đủ tên và ghép với các chữ cái đầu của họ và tên đệm. Nếu có nhiều người cùng email thì từ người thứ 2 sẽ thêm số thứ tự vào email đó.

Cho trước các xâu họ tên (có thể không chuẩn). Hãy tạo ra các địa email tương ứng.

**Input:**

* Dòng 1 ghi số N là xâu họ tên trong danh sách
* N dòng tiếp theo ghi lần lượt các xâu họ tên (không quá 50 ký tự)

**Output:** Ghi ra các email được tạo ra.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4      nGUYEn    quaNG   vInH     tRan   thi THU    huOnG     nGO   quoC  VINH   lE            tuAn    aNH | vinhnq@ptit.edu.vn  huongttt@ptit.edu.vn  vinhnq2@ptit.edu.vn  anhlt@ptit.edu.vn |

**‘**

**#include<stdio.h>**

**#include<string.h>**

**int kt(char s[],char a[]){**

**int l=strlen(s);**

**int n=strlen(a);**

**if(l!=n) return 0;**

**else{**

**for(int i=0;i<l;i++){**

**if(s[i]!=a[i]) return 0;**

**}**

**return 1;**

**}**

**}**

**int main(){**

**int t;**

**char a[101],s[101][101];**

**scanf("%d",&t);**

**int j,k,h;**

**for(k=0;k<t;k++){**

**j=0;**

**scanf("\n");**

**gets(a);**

**int d=0,i,vt,l=strlen(a);**

**for(i=0;i<l;i++){**

**if(a[i]>=65 && a[i]<=90) a[i]=a[i]+32;**

**}**

**while(a[d]==' '){**

**d++;**

**}**

**while(a[l-1]==' '){**

**l--;**

**}**

**for(i=l-1;i>=0;i--){**

**if(a[i]==' '){**

**break;**

**}**

**vt=i;**

**}**

**for(i=vt;i<l;i++){**

**s[k][j++]=a[i];**

**}**

**s[k][j++]=a[d];**

**for(i=d+1;i<vt;i++){**

**if(a[i]!=' ' && a[i-1]==' ') s[k][j++]=a[i];**

**}**

**int dem=1;**

**for(h=0;h<k;h++){**

**if(kt(s[h],s[k])==1) dem++;**

**}**

**printf("%s",s[k]);**

**if(dem>1) printf("%d",dem);**

**printf("@ptit.edu.vn\n");**

**}**

**}**

TÌM TỪ DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06030/top)

Cho một văn bản bất kỳ.  Hãy tìm ra từ có độ dài lớn nhất. Nếu có nhiều từ khác nhau có độ dài bằng nhau và bằng giá trị lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file dữ liệu vào (nhưng một từ dù xuất hiện nhiều lần cũng chỉ được liệt kê một lần).

**Input:**Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

**Output:**Ghi ra từ dài nhất, độ dài của nó và số lần xuất hiện trong file, cách nhau một khoảng trống. Nếu có nhiều từ như vậy thì liệt kê lần lượt các từ theo thứ tự xuất hiện trong file ban đầu.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Tiet hoc cuoi cung da ket thuc. Mon hoc Tin hoc co so 2 da ket thuc. Cac ban co gang on tap tot de thi dat ket qua cao. Chuc cac ban ngay cang gat hai duoc nhieu thanh cong tren con duong da chon | thuc. 5  2  nhieu 5  1  thanh 5  1  duong 5  1 |

**#include<stdio.h>**

**#include<string.h>**

**struct chuoi{**

**char xau[1001];**

**};**

**int main(){**

**struct chuoi s[1001];**

**int n=0,max=0;**

**scanf(" ");**

**while(scanf("%s",s[n].xau)==1){**

**n++;**

**}**

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

**int l=strlen(s[i].xau);**

**if(l>max) max=l;**

**}**

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

**int kt=0,dem=1;**

**int l=strlen(s[i].xau);**

**if(l==max){**

**for(int j=0;j<i;j++){**

**if(strcmp(s[i].xau,s[j].xau)==0){**

**kt=1;**

**break;**

**}**

**}**

**if(kt==0){**

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

**if(strcmp(s[i].xau,s[j].xau)==0) dem++;**

**}**

**printf("%s %d %d\n",s[i].xau,max,dem);**

**}**

**}**

**}**

**}**

XÂU KÝ TỰ ĐẦY ĐỦ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06031/top)

Một xâu được gọi là đầy đủ nếu xóa đi 0 hoặc nhiều hơn các ký tự từ xâu đó ta thu được xâu abcdefghijklmnopqrstuvwxyz

Cho một xâu ký tự chỉ bao gồm các chữ cái viết thường. Hãy tính số ký tự phải chèn thêm ít nhất (vào bất cứ chỗ nào) để có thể chuyển xâu đó sang dạng đầy đủ.

**Input:** Chỉ có một dòng ghi ra xâu ký tự đầu vào. Không quá 50 chữ cái.

**Output:** Ghi ra số kỹ tự ít nhất phải chèn thêm.

**Ví dụ:**

|  |  |
| --- | --- |
| **Test 1** | **Test 2** |
| Input  xyzabcdefghijklmnopqrstuvw    Output  3 | Input  aiemckgobjfndlhp    Output  20 |

#include <stdio.h>

#include <string.h>

int main() {

char s[100];

int f[100];

gets(s);

int max=1;

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

f[i] = 1;

for (int j=i-1; j>=0; j--) {

if (s[i]>s[j]) {

if(f[i]<f[j]+1) f[i]=f[j]+1;

}

}

}

for (int i=0; i<strlen(s); i++)

if (f[i]>=max)

max = f[i];

printf("%d",26-max);

}

GHÉP XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06032/top)

Cho M xâu kí tự. Nhiệm vụ của bạn là hãy ghép các xâu này thành một từ, sao cho từ thu được có thứ tự từ điển nhỏ nhất.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 100).

Mỗi test gồm số nguyên M (M ≤ 9) là số lượng các từ, theo sau là M xâu.

Mỗi xâu có độ dài không vượt quá 10.

**Output:**

Với mỗi test hãy in ra xâu có thứ tự từ điển nhỏ nhất tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  4 acm ptit for students  5 k duz q rc lvraw  3 a bb cc  5 asf asfi asfi afsi okj  5 ukuy hopji lie jaa dcyi | acmforptitstudents  duzklvrawqrc  abbcc  afsiasfasfiasfiokj  dcyihopjijaalieukuy |

#include<stdio.h>

#include<string.h>

int main() {

int t;

scanf("%d",&t);

while(t--) {

int n;

char ds1[100], ds2[100],tmp[100];

char s[100][100];

scanf("%d",&n);

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

scanf("%s", s[i]);

}

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

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

strcpy(ds1,s[i]);

strcat(ds1,s[j]);

strcpy(ds2,s[j]);

strcat(ds2,s[i]);

if (strcmp(ds1,ds2)>0) {

strcpy(tmp,s[i]);

strcpy(s[i],s[j]);

strcpy(s[j],tmp);

}

}

}

for(int i=0; i<n; i++)printf("%s", s[i]);

printf("\n");

}

}

TRỘN XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06033/top)

Cho hai xâu ký tự S1 và S2 với độ dài N và chỉ chứa các ký tự từ A đến H. Chúng ta thực hiện thao tác như sau:

* Bước đầu tiên tạo S12 bằng cách lấy các ký tự lần lượt trong S1 và S2 từ trái sang phải, lấy một ký tự trong S2 trước, sau đó đến 1 ký tự trong S1 và cứ như vậy. Ví dụ S1 =  ABCHAD, S2= DEFDAC thì S12 = DAEBFCDHAACD
* Sau đó ta lại lấy nữa bên trái của S12 thành S1 mới, nửa bên phải thành S2 mới. Trong ví dụ trên S1 mới là DAEBFC, S2 mới là DHAACD. Rồi lại tiếp tục như vậy trong các bước tiếp theo.

Cho trước một xâu S có độ dài 2\*N. Bài toán đặt ra là liệu có thể tạo ra xâu S sau một số lần lặp hay không.

**Dữ liệu vào**

Có nhiều bộ test, mỗi bộ test có bốn dòng. Dòng đầu ghi số N không quá 200. Dòng thứ 2 ghi S1, dòng thứ 3 ghi S2. Dòng cuối ghi xâu S. Input kết thúc với một dòng ghi số 0.

**Kết quả**

Ghi ra số bước lặp cần thiết. Nếu không thể tìm được thì ghi ra -1.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  AHAH  HAHA  HHAAAAHH  3  CDE  CDE  EEDDCC  0 | 2  -1 |

#include<stdio.h>

#include<string.h>

int main(){

while(-1){

int n;

scanf("%d",&n);

if(n==0) break;

char a[101], b[101], c[202];

scanf("\n");

gets(a);

gets(b);

gets(c);

char s1[101], s2[101];

int i,j,kt=0;

strcpy(s1,a);

strcpy(s2,b);

int d=0;

while(-1){

int m=0;

char s12[1000]="";

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

s12[m++]=s2[i];

s12[m++]=s1[i];

}

d++;

if(strcmp(s12,c)==0) break;

m=0;

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

s1[i]=s12[m++];

}

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

s2[i]=s12[m++];

}

if(strcmp(s1,a)==0 && strcmp(s2,b)==0){

kt=1;

break;

}

}

if(kt==1) printf("-1\n");

else printf("%d\n",d);

}

}

SỐ LA MÃ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06034/top)

Bảng chữ số La Mã bao gồm các chữ cái với ý nghĩa I=1; V=5; X=10; L=50; C=100;D=500; M=1000. Một số quy tắc viết các số La Mã như sau:

* Tính từ trái sang phải giá trị của các chữ số và nhóm chữ số giảm dần.
* I chỉ có thể đứng trước V hoặc X, X chỉ có thể đứng trước L hoặc C, C chỉ có thể đứng trước D hoặc M.
* Các chữ cái I, X, C, M, không được lặp lại quá ba lần liên tiếp; các chữ cái V, L, D không được lặp lại quá một lần liên tiếp.

Bài toán đặt ra là cho một xâu ký tự mô tả **đúng** một số La Mã. Hãy tính giá trị thập phân của số đó

**Input:**Dòng đầu ghi số bộ test. Mỗi bộ test ghi trên một dòng dãy ký tự số La Mã.

**Output:** Với mỗi bộ test ghi ra kết quả tương ứng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  XIX  DC  CD | 19  600  400 |

#include<stdio.h>

#include<string.h>

int main(){

int t;

int b[10000];

char a[100];

scanf("%d",&t);

while(t--){

scanf("\n");

gets(a);

int i,l=strlen(a);

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

if(a[i]=='I') b[i]=1;

else if(a[i]=='V') b[i]=5;

else if(a[i]=='X') b[i]=10;

else if(a[i]=='L') b[i]=50;

else if(a[i]=='C') b[i]=100;

else if(a[i]=='D') b[i]=500;

else b[i]=1000;

}

int c=b[l-1];

for(i=0;i<l-1;i++){

if(b[i]>=b[i+1]) c=c+b[i];

else c=c-b[i];

}

printf("%d\n",c);

}

}

TỔNG HAI SỐ NGUYÊN LỚN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06024/top)

Viết chương trình cộng hai số nguyên dương bất kỳ (không quá 500 chữ số).  
**Input:**  
Dòng 1 ghi số bộ test. Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một số nguyên  
dương  
**Output:**  
Với mỗi bộ test ghi ra một số nguyên dương là tổng hai số đã cho (số này cũng  
không quá 500 chữ số).  
**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 12 100 1212 8888 121212121212121212 45678978 | 112 10100 121212121257800190 |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

char a[501],b[501],c[501];

while(t--){

scanf("\n");

gets(a);

scanf("\n");

gets(b);

int m=strlen(a),n=strlen(b);

int l1=m,l2=n,i;

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

a[i]-=48;

}

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

b[i]-=48;

}

int j=0,e=0,d;

while(-1){

d=a[m-1]+b[n-1]+e;

c[j]=d%10;

e=d/10;

if(l1>l2 && n-1==0) break;

if(l2>l1 && m-1==0) break;

if(l1==l2 && n-1==0){

if(e>0){

j++;

c[j]=e;

}

break;

}

m--;

n--;

j++;

}

if(l1>l2){

while(m-1>0){

m--;

j++;

d=a[m-1]%10+e;

c[j]=d%10;

e=d/10;

if(m-1==0){

if(e>0){

j++;

c[j]=e;

}

}

}

}

if(l2>l1){

while(n-1>0){

n--;

j++;

d=b[n-1]%10+e;

c[j]=d%10;

e=d/10;

if(n-1==0){

if(e>0){

j++;

c[j]=e;

}

}

}

}

for(int k=0;k<=j;k++){

printf("%c",c[j-k]+48);

}

printf("\n");

}

}

TỔNG ĐA THỨC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07008/top)

Cho hai đa thức có bậc không quá 10000 (chỉ viết ra các phần tử có hệ số khác 0). Hãy tính tổng hai đa thức đó.

**Input:** Dòng đầu ghi số bộ test. Mỗi bộ test có hai dòng, mỗi dòng ghi một đa thức theo mẫu như trong ví dụ. Chú ý: Bậc của các hạng tử luôn theo thứ tự giảm dần, trong đa thức chỉ có phép cộng và luôn được viết đầy đủ hệ số + số mũ (kể cả mũ 0).

**Output:** Ghi ra một dòng đa thức tổng tính được (theo mẫu như ví dụ)

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  3\*x^8 + 7\*x^2 + 4\*x^0  11\*x^6 + 9\*x^2 + 2\*x^1 + 3\*x^0 | 3\*x^8 + 11\*x^6 + 16\*x^2 + 2\*x^1 + 7\*x^0 |

#include<stdio.h>

struct pt{

int hs;

int mu;

};

int main(){

int t;

scanf("%d",&t);

while(t--){

struct pt a[10000],d;

int x=0,y=0;

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

scanf("%d",&a[i].hs);

scanf("\*x^");

scanf("%d",&a[i].mu);

x=i+1;

if(a[i].mu==0)break;

scanf(" + ");

}

for(int i=x;;i++){

scanf("%d",&a[i].hs);

scanf("\*x^");

scanf("%d",&a[i].mu);

y=i+1;

if(a[i].mu==0)break;

scanf(" + ");

}

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

for(int j=i+1;j<y;j++){

if(a[i].mu<a[j].mu) {

d=a[j];

a[j]=a[i];

a[i]=d;

}

}

}

for(int i=0;i>=0;i++){

if(a[i].mu!=a[i+1].mu){

printf("%d",a[i].hs);

printf("\*x^");

printf("%d",a[i].mu);

if(a[i].mu==0)break;

printf(" + ");

}

if(a[i].mu==a[i+1].mu){

a[i].hs=a[i].hs +a[i+1].hs;

printf("%d",a[i].hs);

printf("\*x^");

printf("%d",a[i].mu);

i++;

if(a[i+1].mu==0)break;

printf(" + ");

}

}

printf("\n");

}

}

Ố CHỈ CÓ BA ƯỚC SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03051/top)

Cho hai số L, R. Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số trong khoảng [L, R]. Ví dụ L =1, R =10, ta có kết quả là 2 vì chỉ có số 4 và 9 là có đúng 3 ước số.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số L, R.
* T, N thỏa mãn rang buộc 1≤T≤100; 1≤L, R ≤1012.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1 10  1 1000000000000 | 2  78498 |

#include<stdio.h>

#include<math.h>

int b[1000000+9]={0};

void sangnt(){

int i,j;

for(i=2;i<=1000000;i++){

if(b[i]==0){

for(j=i\*2;j<=1000000;j=j+i){

b[j]=1;

}

}

}

}

int main(){

int t;

scanf("%d",&t);

sangnt();

long s=0;

while(t--){

long long m,n;

scanf("%lld %lld",&m,&n);

long m1=sqrt(m),n1=sqrt(n),i,dem=0;

for(i=m1;i<=n1;i++){

if(b[i]==0 && i>1) dem++;

}

printf("%ld\n",dem);

}

}

ƯỚC SỐ CỦA GIAI THỪA

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03038/top)

Cho số tự nhiên N và số nguyên tố P. Nhiệm vụ của bạn là tìm số x lớn nhất để N! chia hết cho px. Ví dụ với N=7, p=3 thì x=2 là số lớn nhất để 7! Chia hết cho 32.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số N, p được viết cách nhau một vài khoảng trống.
* T, N, p thỏa mãn ràng buộc : 1≤T≤100; 1≤N≤105; 2≤p≤5000;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 3 62  7 76  2 3  5 | 9 73 0 |

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

int a,b,i,d=0,k;

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

for(i=1;i<=a;i++){

k=i;

while(k>=b && k%b==0){

d++;

k=k/b;

}

}

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

}

}

BỘI SỐ NHỎ NHẤT CỦA N SỐ NGUYÊN DƯƠNG ĐẦU TIÊN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03058/top)

Cho số tự nhiên n. Nhiệm vụ của bạn là tìm số nguyên nhỏ nhất chia hết cho 1, 2, .., n.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* T dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một số tự nhiên n.
* T thỏa mãn ràng buộc: 1≤T≤104; n không quá 100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  3  5 | 6  60 |

#include<stdio.h>

#include<math.h>

int b[1000+9]={0};

int c[1009];

int main(){

int t;

scanf("%d",&t);

while(t--){

int n,i,j,k;

scanf("%d",&n);

long long s=1,x;

for(i=2;i<=n;i++){

k=i;

for(j=2;j<=sqrt(k);j++){

int dem=0;

x=1;

if(k%j==0){

while(k%j==0){

x=x\*j;

dem++;

k=k/j;

}

if(dem>b[j]){

b[j]=dem;

c[j]=x;

}

}

}

if(k>1 && b[k]==0){

b[k]++;

c[k]=k;

}

}

for(int i=2;i<=n;i++){

if(b[i]>0) s=s\*c[i];

}

for(int i=2;i<=n;i++){

b[i]=0;

c[i]=0;

}

printf("%lld",s);

printf("\n");

}

}

BỘI SỐ NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03062/top)

Cho hai số tự nhiên n, m . Nhiệm vụ của bạn là tìm số nguyên nhỏ nhất chia hết cho n , n+1, .., m.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* T dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là 2 số tự nhiên n, m.
* T thỏa mãn ràng buộc: 1≤T≤104; 1≤n, m ≤40

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1 3  3 5 | 6  60 |

#include<stdio.h>

#include<math.h>

int b[1000+9]={0};

int c[1009];

int main(){

int t;

scanf("%d",&t);

while(t--){

int n,m,i,j,k;

scanf("%d %d",&n,&m);

long long s=1,x;

for(i=n;i<=m;i++){

k=i;

for(j=2;j<=sqrt(k);j++){

int dem=0;

x=1;

if(k%j==0){

while(k%j==0){

x=x\*j;

dem++;

k=k/j;

}

if(dem>b[j]){

b[j]=dem;

c[j]=x;

}

}

}

if(k>1 && b[k]==0){

b[k]++;

c[k]=k;

}

}

for(int i=2;i<=m;i++){

if(b[i]>0) s=s\*c[i];

}

for(int i=2;i<=m;i++){

b[i]=0;

c[i]=0;

}

printf("%lld",s);

printf("\n");

}

}

SỐ ĐẦU TIÊN BỊ LẶP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01013/top)

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy tìm số xuất hiện nhiều hơn 1 lần trong dãy số và số thứ tự là nhỏ nhất.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test gồm số nguyên N (1≤ N ≤ 100 000), số lượng phần tử trong dãy số ban đầu.

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 10^9).

**Output:**

Với mỗi test in ra đáp án của bài toán trên một dòng. Nếu không tìm được đáp án, in ra “NO”.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  7  10 5 3 4 3 5 6  4  1 2 3 4 | 5  NO |

#include<iostream>

using namespace std;

int main(){

int t,k;

cin>>t;

for(k=0;k<t;k++){

long n,j,i,kt=0,x;

cin>>n;

long a[n];

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

cin>>a[i];

}

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

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

if(a[i]==a[j]){

kt++;

}

}

if(kt!=0) {

cout<<a[i];

break;

}

}

if(kt==0) {

cout<<"NO";

}

cout<<endl;

}

}

TÌM SỐ DƯ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01015/top)

Hãy tính giá trị biểu thức .

**Input:**

Dòng đầu là số lượng bộ test T (T ≤ 100).

Mỗi test gồm một xâu biểu diễn số nguyên n, n có không quá 100 000 kí tự.

**Output:**

Với mỗi test in ra đáp án tìm được trên một dòng.

**Test ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  4  123456789 | 4  0 |

#include<stdio.h>

#include<math.h>

#include<string.h>

int main(){

int t,k;

scanf("%d",&t);

char a[100000];

for(k=0;k<t;k++){

scanf("\n");

gets(a);

long l=strlen(a),i,d=0;

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

d=10\*d+a[i];

}

if(d%4==0) printf("4\n");

else printf("0\n");

}

}

ƯỚC SỐ CHUNG LỚN NHẤT (GCD)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01016/top)

Cho dãy số A[] nguyên dương có N phần tử. Bạn cần xây dựng dãy số B có N+1 phần tử sao cho gcd(B[i], B[i+1]) = A[i] với mọi i thỏa mãn 1 <= i <= n. Vì có rất nhiều dãy số B[] thỏa mãn, nên bạn cần tìm được dãy số có tổng các phần tử là nhỏ nhất.

**Input**

* Dòng đầu tiên là số lượng bộ test T (1 <= T <= 10).
* Mỗi test bắt đầu bằng số nguyên N (2 <= N <= 1000).
* Dòng tiếp theo gồm N số nguyên A[i] (1 <= A[i] <= 10 000).

**Output**

* Với mỗi test in ra dãy số B[] trên một dòng.

**Test ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  3  1 2 3  3  5 10 5 | 1 2 3 6  5 10 10 5 |

#include<iostream>

#include<math.h>

using namespace std;

int ucll(long x, long y){

long d;

while(x%y!=0){

d=x%y;

x=y;

y=d;

}

return y;

}

int bcnn(long x, long y){

long z=x\*y/ucll(x,y);

return z;

}

int main(){

int t,k;

cin>>t;

for(k=0;k<t;k++){

int n,i

;

cin>>n;

long a[n],b[n+1];

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

cin>>a[i];

}

cout<<a[1]<<" ";

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

b[i]=bcnn(a[i],a[i+1]);

cout<<b[i]<<" ";

}

cout<<a[n];

cout<<endl;

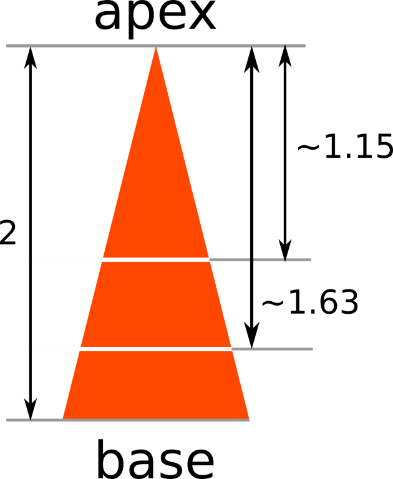
}

}

CHIA TAM GIÁC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01017/top)

Cho một tam giác cân có độ dài đáy bằng 1 và chiều cao bằng H. Bạn cần chia tam giác này thành N phần có diện tích bằng nhau và song song với đáy.



Hãy in ra N-1 số là khoảng cách từ lát cắt thứ i tới đỉnh của tam giác.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 100).

Mỗi test gồm 2 số nguyên N và H (2 ≤ ≤ N ≤ 1000, 1 ≤ H ≤ 100 000).

**Output:**

Với mỗi test, in ra đáp án tìm được trên một dòng với 6 chữ số sau dấu phảy.

|  |  |
| --- | --- |
| Input: | Output |
| 2  3 2  2 100000 | 1.154701 1.632993  70710.678119 |

#include<stdio.h>

#include<math.h>

int main(){

int t,o;

scanf("%d",&t);

for(o=0;o<t;o++){

int n,h,i;

scanf("%d %d",&n,&h);

double s,s1=0,h1,k,m;

s=(double)h/2;

m=s/n;

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

s1=s1+m;

k=s1/s;

h1=h\*sqrt(k);

printf("%lf ",h1);

}

printf("\n");

}

}

DI CHUYỂN ROBOT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01025/top)

Một robot xuất phát từ vị trí (0,0) mặt quay về hướng Bắc. Mỗi lần chỉ có một trong 4 lệnh chuyển động là G, L, R, B tương ứng là tiến về phía trước, tiến sang trái, tiến sang phải, lùi lại phía sau một đơn vị.

Cho dãy lệnh chuyển động. Hãy tính xem vị trí cuối cùng của robot là vị trí nào?

**Input**

* Dòng đầu tiên ghi n (n≤100) là số lệnh robot cần thực hiện.
* Dòng thứ hai là dãy n ký tự mô tả dãy lệnh robot thực hiện

**Output**

Ghi ra hai số nguyên là tọa độ (x,y) của vị trí cuối cùng robot

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  GLLRB | -1 0 |
| **Input** | **Output** |
| 2  RG | -2 0 |

#include<stdio.h>

#include<string.h>

int main(){

int n;

scanf("%d",&n);

char a[n];

scanf("\n");

gets(a);

int x=0,y=0;

int k=2;

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

if(a[i]==71){

if(k==1){

x=x+1; k=1; }

else if(k==2){

y=y+1; k=2; }

else if(k==3){

x=x-1; k=3;}

else {

y=y-1; k=4;}

}

else if(a[i]==76){

if(k==1) {

y=y+1; k=2; }

else if(k==2) {

x=x-1; k=3; }

else if(k==3) {

y=y-1; k=4; }

else {

x=x+1; k=1;}

}

else if(a[i]==82){

if(k==1){

y=y-1; k=4; }

else if(k==2){

x=x+1; k=1;}

else if(k==3){

y=y+1; k=2;}

else {

x=x-1; k=3; }

}

else {

if(k==1){

x=x-1; k=3; }

else if(k==2){

y=y-1; k=4; }

else if(k==3) {

x=x+1; k=1; }

else {y+=1; k=2; }

}

}

printf("%d %d",x,y);

return 0;

}

BỐC BI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01029/top)

Unnie có một hộp bi trong đó có a viên bi màu đỏ, b viên bi màu xanh và c viên bi màu vàng. Unnie đố Oppa có thể nhắm mắt mà lấy ra được k viên bi có cùng màu.

Oppa vừa đi dép lê vừa suy nghĩ nhưng mà không biết phải làm như thế nào. Nhờ các bạn tính giúp xem Oppa cần lấy ít nhất bao nhiêu viên để đảm bảo yêu cầu của Unnie nha.

**Input**

Chứa 4 số nguyên a, b, c, k. (1 ≤ a, b, c ≤ 1000, 1 ≤ k ≤ max(a, b, c))

**Output**

In ra 1 số nguyên là đáp án của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4 6 3 | 7 |

#include<stdio.h>

int main(){

int a,b,c,k;

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

if(a>=k && b>=k && c>=k) printf("%d",3\*k-2);

else if(a<k && b<k && c>=k ) printf("%d",a+b+k);

else if( a<k && b>=k && c<k ) printf("%d",a+c+k);

else if (a>=k && b<k && c<k) printf("%d",b+c+k);

else if(a<k && b>=k && c>=k )printf("%d",a+k+k-1);

else if(a>=k && b<k && c>=k) printf("%d",b+k+k-1);

else printf("%d",c+k+k-1);

}

PHẦN TỬ LỚN NHẤT TRONG DÃY CON

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01030/top)

Cho dãy số A[] gồm có N phần tử và số nguyên K.

Với mỗi dãy con liên tiếp có độ dài bằng K (từ trái sang phải), bạn hãy in ra phần tử lớn nhất trong dãy con này.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test gồm số nguyên N và K (1≤ N ≤ 100 000, 1 ≤ K ≤ N).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 109).

**Output:**

Với mỗi test, in ra trên một dòng N-K+1 số nguyên là đáp án tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  9 3  1 2 3 1 4 5 2 3 6  10 4  8 5 10 7 9 4 15 12 90 13 | 3 3 4 5 5 5 6  10 10 10 15 15 90 90 |

#include<iostream>

#include<math.h>

#include<algorithm>

using namespace std;

int main(){

int t,o;

cin>>t;

for(o=0;o<t;o++){

int n,k,i,j;

cin>>n>>k;

long a[n],max;

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

cin>>a[i];

}

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

max=0;

for(j=i;j<i+k;j++){

if(a[j]>max){

max=a[j];

}

}

cout<<max<<" ";

}

cout<<endl;

}

}

SỐ XUẤT HIỆN NHIỀU NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01004/top)

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy tìm một số có tần số xuất hiện nhiều nhất, yêu cầu lớn hơn N/2 lần xuất hiện trong dãy số.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test gồm số nguyên N (1≤ N ≤ 100 000), số lượng phần tử trong dãy số ban đầu.

Dòng tiếp theo gồm N số nguyên A[i] (1 ≤ A[i] ≤ 1 000 000).

**Output:**

Với mỗi test in ra đáp án của bài toán trên một dòng. Nếu không tìm được đáp án, in ra “NO”.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  9  3 3 4 2 4 4 2 4 4  8  3 3 4 2 4 4 2 4 | 4  NO |

#include<stdio.h>

#include<math.h>

int main(){

int n;

scanf("%d",&n);

while(n--){

long m;

scanf("%ld",&m);

long a[m];

for(long i=0;i<m;i++){

scanf("%ld",&a[i]);

}

long x;

long y=m/2;

long kt=0;

for(long i=0;i<=y;i++){

long dem=1;

for(long j=i+1;j<m;j++){

if(a[j]==a[i]){

dem++;

}

}

if(dem>y) {

x=a[i];

kt=1;

break;

}

}

if(kt==1)printf("%ld",x);

if(kt==0)printf("NO");

printf("\n");

}

}

TẦN SUẤT LẺ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01005/top)

Cho dãy số A[] gồm có N phần tử. Các phần tử trong dãy số đều xuất hiện với tần suất chẵn, chỉ có duy nhất 1 số có số lần xuất hiện là số lẻ. Nhiệm vụ của bạn là hãy tìm số này.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test gồm số nguyên N (1≤ N ≤ 100 000), số lượng phần tử trong dãy số ban đầu. N là một số lẻ.

Dòng tiếp theo gồm N số nguyên A[i] (1 ≤ A[i] ≤ 1 000 000).

**Output:**

Với mỗi test in ra trên mỗi dòng một số nguyên là đáp án của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  7  1 2 3 2 3 1 3  5  1 1 3 3 2 | 3  2 |

#include<iostream>

int b[1000001];

using namespace std;

int main(){

int t,k;

cin>>t;

for(k=0;k<t;k++){

long n,i,x;

cin>>n;

long a[n];

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

cin>>a[i];

b[a[i]]++;

}

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

if(b[a[i]]%2==1) x=a[i];

}

cout<<x<<endl;

}

}

 QUAY VÒNG DÃY SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CP01008/top)

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy chuyển D phần tử đầu tiên của dãy A[] xuống cuối dãy.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test gồm số nguyên N và D (1≤ N ≤ 1000, 0 ≤ D ≤ N).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 10^6).

**Output:**

Với mỗi test, in ra trên một dòng là dãy số thu được sau khi thực hiện phép quay vòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 1  7 2  1 2 3 4 5 6 7 | 3 4 5 6 7 1 2 |

#include<iostream>

using namespace std;

int main(){

int t,k;

cin>>t;

for(k=0;k<t;k++){

int n,d,j,i;

cin>>n>>d;

long a[n];

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

cin>>a[i];

}

for(i=d;i<n;i++){

cout<<a[i]<<" ";

}

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

cout<<a[i]<<" ";

}

cout<<endl;

}

}