ĐẾM SỐ PHẦN TỬ LỚN HƠN SỐ ĐỨNG TRƯỚC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04011/top)

Cho một dãy số nguyên dương có n phần tử (2<=n<=50). Hãy liệt kê số các phần tử trong dãy không nhỏ hơn các số đứng trước nó (tính cả phần tử đầu tiên).

**Input:** Dòng 1 ghi số bộ test. Mỗi bộ test ghi ra 2 dòng: dòng 1 là số phần tử, dòng thứ 2 ghi dãy số**.**

**Output:** ghi ra số phần tử thỏa mãn.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  7  3 5 6 8 4 2 9 | 5 |

#include <stdio.h>

int main(){

int t,k;

scanf("%d",&t);

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

int n,i;

scanf("%d", &n);

int a[n];

int d=0;

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

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

}

int max=a[0];

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

if(a[i]>= max){

d++;

max=a[i];

}

}

printf("%d",d);

printf("\n");

}

return 0;

}

ĐẾM SỐ LẦN XUẤT HIỆN 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04015/top)

Cho dãy số A có n phần tử chỉ bao gồm các số nguyên dương (không quá 105). Hãy đếm xem mỗi số xuất hiện bao nhiêu lần.

**Input:**Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: dòng đầu ghi số n (không quá 100); dòng tiếp theo ghi n số của dãy.

**Output:**Với mỗi bộ test ghi ra **thứ tự bộ test**, sau đó lần lượt là các số nguyên tố trong dãy **theo thứ tự xuất hiện trong dãy** và số lần xuất hiện của nó.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  10  1 7 2 8 3 3 2 1 3 2 | Test 1:  1 xuat hien 2 lan  7 xuat hien 1 lan  2 xuat hien 3 lan  8 xuat hien 1 lan  3 xuat hien 3 lan |

#include<stdio.h>

int b[10001];

int main(){

int t,k;

scanf("%d",&t);

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

int n,i,d=0,j;

scanf("%d",&n);

int a[n];

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

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

b[a[i]]++;

}

printf("Test %d:\n",k+1);

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

int kt=0;

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

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

kt=1;

break;

}

}

if(kt==0) printf("%d xuat hien %d lan\n",a[i],b[a[i]]);

}

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

b[a[i]]=0;

}

printf("\n");

}

}

ĐẾM CÁC SỐ NGUYÊN TỐ TRONG DÃY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04016/top)

Cho dãy số A có n phần tử chỉ bao gồm các số nguyên dương (không quá 105). Hãy xác định các số nguyên tố trong dãy và đếm xem mỗi số xuất hiện bao nhiêu lần.

**Input:**Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: dòng đầu ghi số n (không quá 100); dòng tiếp theo ghi n số của dãy.

**Output:**Với mỗi bộ test ghi ra thứ tự bộ test, sau đó lần lượt là các số nguyên tố trong dãy **theo thứ tự từ nhỏ đến lớn** và số lần xuất hiện của nó.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  10  1 7 2 8 3 3 2 1 3 2 | Test 1:  2 xuat hien 3 lan  3 xuat hien 3 lan  7 xuat hien 1 lan |

#include<stdio.h>

#include<math.h>

int b[10001];

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

int t,k;

scanf("%d",&t);

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

int n,i,d=0,j;

scanf("%d",&n);

int a[n];

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

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

if(nt(a[i])==1) b[a[i]]++;

}

printf("Test %d:\n",k+1);

int kt;

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

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

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

int z=a[j];

a[j]=a[i];

a[i]=z;

}

}

}

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

if(nt(a[i])==1){

kt=0;

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

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

kt=1;

break;

}

}

if(kt==0) printf("%d xuat hien %d lan\n",a[i],b[a[i]]);

}

}

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

b[a[i]]=0;

}

printf("\n");

}

}

SỐ CẶP BẰNG NHAU TRONG DÃY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04018/top)

Viết chương trình đếm các cặp số bằng nhau liên tiếp trong dãy số nguyên.

**Input:**Dòng đầu tiên ghi số bộ test. Mỗi bộ test có hai dòng:

* Dòng đầu ghi số phần tử của dãy, không quá 30
* Dòng tiếp theo ghi các phần tử của dãy, mỗi phần tử cách nhau một khoảng trống. Các phần tử không quá 100.

**Output:**Mỗi bộ test viết ra trên một dòng giá trị tổng chữ số tương ứng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  1  3  3  4  12  1  2  3  3  3  3  4  4  5  5  5  1 | 1  6 |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

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

int n,i,d=0,j;

scanf("%d",&n);

int a[n];

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

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

}

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

if(a[i]==a[i+1]) d++;

}

printf("%d",d);

printf("\n");

}

}

MA TRẬN CHUYỂN VỊ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05001/top)

Viết chương trình nhập một ma trận số nguyên dương cỡ M\*N với 2<N,M<10, các số không quá 100. Hãy in ra chuyển vị của ma trận đó.

**Dữ liệu vào**

Mỗi bộ test viết trên một dòng hai số N và M lần lượt là số hàng và số cột của ma trận (2<N,M<10). Tiếp thep là N dòng ghi các số của ma trận.

**Kết quả**

            In ra ma trận chuyển vị tương ứng.

**Ví dụ**

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

#include<stdio.h>

int main(){

int n,m,i,j;

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

int a[m][n];

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

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

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

}

}

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

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

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

}

printf("\n");

}

}

LOẠI BỎ HÀNG VÀ CỘT ĐẦU TIÊN RA KHỎI MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05005/top)

Cho một ma trận cấp M\*N chỉ bao gồm các số nguyên dương không quá 1000. Hãy viết chương trình trước hết loại bỏ hàng đầu tiên sau đó loại bỏ cột đầu tiên ra khỏi ma trận.

**Input:**Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số N và M lần lượt là số hàng và số cột của ma trận (2<N,M<10). Tiếp thep là N dòng ghi các số của ma trận.

**Output:**Mỗi bộ test viết ra thứ tự bộ test, sau đó là ma trận sau khi xử lý.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 3 3 1 2 4 3 4 0 6 3 5 | Test 1:  4 0 3 5 |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

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

int n,m,i,j;

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

int a[m][n];

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

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

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

}

}

printf("Test %d:\n",k+1);

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

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

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

}

printf("\n");

}

}

printf("\n");

}

ĐẾM SỐ TỪ TRONG XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06003/top)

Một từ trong xâu ký tự được định nghĩa là một dãy ký tự liên tiếp không có khoảng trống. Hãy đếm số từ của xâu và in kết quả ra màn hình.

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi bộ test có một xâu không quá 200 ký tự.

**Kết quả:** Ghi ra số từ đếm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Lap trinh     C va C++  ACBDSDS kdfjdkgfdgkhfgjlfgdkjfdgdgfdgfd | 5  2 |

#include<stdio.h>

#include<string.h>

int main(){

int t,k;

char a[1000];

scanf("%d",&t);

scanf("\n");

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

gets(a);

int d;

if(a[0]==' ') d=0;

else d=1;

int l=strlen(a);

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

l--;

}

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

if(a[i]==' ' && a[i+1]!=' ') d++;

}

printf("%d",d);

printf("\n");

}

}

SỐ ĐẸP 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06010/top)

Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số chẵn. Viết chương trình đọc vào các số nguyên dương có không quá 500 chữ số và kiếm tra xem số đó có đẹp hay không.

**Input:**

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương n không quá 500 chữ số.

**Output:**

            Mỗi bộ test viết ra trên một dòng chữ YES nếu đó là số đẹp, chữ NO nếu ngược lại

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  123456787654321  86442824468  8006000444422220000222244440006008  235365789787654324567856578654356786556 | NO  YES  YES  NO |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

char a[501];

gets(a);

int i,l=strlen(a);

int n=l/2;

int kt=1;

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

if(a[i]%2!=0 || a[l-1-i]%2!=0 || a[i]!=a[l-1-i]){

printf("NO");

kt=0;

break;

}

}

if(kt==1) printf("YES");

printf("\n");

}

}

SỐ ĐẸP 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06011/top)

Một số được coi là đẹp nếu đó là số thuận nghịch, bắt đầu và kết thúc bằng chữ số 8 và tổng chữ số chia hết cho 10. Viết chương trình đọc vào các số nguyên dương có không quá 500 chữ số và kiếm tra xem số đó có đẹp hay không.

**Input:**

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương n không quá 500 chữ số.

**Output:**

            Mỗi bộ test viết ra trên một dòng chữ YES nếu đó là số đẹp, chữ NO nếu ngược lại

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  123456787654321  8644281154664511824468  8006000444400000000000044440006008  82123400000000000000000000000432128 | NO  NO  YES  YES |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

char m[501];

gets(m);

int i,l=strlen(m);

int n=(l-1)/2;

int kt=1;

int s=0;

int a[strlen(m)];

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

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

s=s+a[i];

}

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

if(a[0]!=8 || a[l-1]!=8 || a[i]!=a[l-1-i]||s%10!=0){

printf("NO");

kt=0;

break;

}

}

if(kt==1 ) printf("YES");

printf("\n");

}

}

SỐ ĐẸP 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06012/top)

Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số nguyên tố. Viết chương trình đọc vào các số nguyên dương có không quá 500 chữ số và kiếm tra xem số đó có đẹp hay không.

**Input:**

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương n không quá 500 chữ số.

**Output:**

            Mỗi bộ test viết ra trên một dòng chữ YES nếu đó là số đẹp, chữ NO nếu ngược lại

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  123456787654321  235755557532  2222333355557777235775327777555533332222 | NO  YES  YES |

#include<stdio.h>

#include<string.h>

#include<math.h>

int b[10001];

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

char m[501];

gets(m);

int i,l=strlen(m);

int n=(l-1)/2;

int kt=1,d=1;

int s=0;

int a[strlen(m)];

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

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

if(nt(a[i])==0){

d=0;

break;

}

}

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

else{

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

if( a[i]!=a[l-1-i]){

printf("NO");

kt=0;

break;

}

}

if(kt==1 ) printf("YES");

}

printf("\n");

}

}

 TÁCH TỪ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06001/top)

Nhập xâu ký tự S có độ dài không quá 100. Một từ được định nghĩa là một dãy ký tự không có khoảng trống.

Hãy tách xâu S thành các từ, mỗi từ in trên một dòng.

**Input:**

Chỉ có một dòng ghi xâu S (độ dài không quá 100)

**Output:**

Ghi ra kết quả.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Tin hoc co so 2 | Tin  hoc  co  so  2 |

#include<stdio.h>

#include<string.h>

int main(){

char a[100];

gets(a);

int l=strlen(a);

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

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

a[i]='\n';

}

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

}

}

ƯỚC SỐ CHUNG LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01027/top)

Viết chương trình tính ước số chung lớn nhất của 2 số nguyên dương (không quá 6 chữ số).

**Input**

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương.

**Output**

Mỗi bộ test ghi ra kết quả tính được trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  24 14  75 125 | 2  25 |

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

int a,b;

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

while(a%b!=0){

int d=a%b;

a=b;

b=d;

}

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

}

}

SỐ CHẴN ĐẶC BIỆT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03029/top)

Một số gọi là số chẵn đặc biệt nếu nó là số chẵn và tất cả các chữ số đều chẵn. Viết chương trình kiểm tra xem số đã cho có phải là số chẵn đặc biệt hay không.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi số N không quá 18 chữ số

**Output**

* Với mỗi bộ test, nếu N là số chẵn đặc biệt thì ghi ra YES, ngược lại ghi ra NO trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  123456  22446688000000 | NO  YES |

#include<stdio.h>

int kt(long long a){

if(a%2==1) return 0;

else {

while(a>0){

int d=a%10;

if(d%2==1) return 0;

a=a/10;

}

return 1;

}

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long long n;

scanf("%lld",&n);

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

else printf("NO\n");

}

}

 SỐ KHÔNG GIẢM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03030/top)

Một số gọi là số không giảm nếu các chữ số từ trái qua phải tạo thành dãy không giảm (tức là không có chữ số nào phía sau nhỏ hơn chữ số ngay trước nó). Viết chương trình liệt kê các số không giảm có N chữ số (1<N<7).

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi số N

**Output**

* Với mỗi bộ test, ghi ra lần lượt các số không giảm có N chữ số, các số cách nhau một khoảng trống.
* Hết một bộ test thì xuống dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2 | 11 12 13 14 15 16 17 18 19 22 23 24 25 26 27 28 29 33 34 35 36 37 38 39 44 45 46 47 48 49 55 56 57 58 59 66 67 68 69 77 78 79 88 89 99 |

#include<stdio.h>

int nt(long a){

int e=a%10;

a=a/10;

while(a>0){

int d=a%10;

if(e<d) return 0;

e=d;

a/=10;

}

return 1;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long n;

scanf("%ld",&n);

long x=1,y,i;

while(n--){

x=x\*10;

}

y=x/10;

for(i=y;i<x;i++){

if(nt(i)==1) printf("%ld ",i);

}

printf("\n");

}

}

 CẶP SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03031/top)

Cho hai cặp số (a,b) và (c,d). Hãy tính toán xem ước số chung lớn nhất của hai cặp số này có bằng nhau hay không.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi số 4 số a, b, c, d. Các số không quá 109

**Output**

* Với mỗi bộ test, ghi ra YES nếu 2 cặp số (a,b) và (c,d) có ước chung lớn nhất bằng nhau, ngược lại ghi ra NO trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2 4 4 8  3 5 7 9 | NO  YES |

#include<stdio.h>

long ucln(long a, long b){

long d;

while(a%b!=0){

d=a%b;

a=b;

b=d;

}

return b;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long a,b,c,d;

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

if(ucln(a,b)==ucln(c,d)) printf("YES");

else printf("NO");

printf("\n");

}

}

ĐẾM SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03032/top)

Viết chương trình đếm xem trong đoạn [a,b] có bao nhiêu số là số nguyên tố và tất cả các chữ số của nó cũng là số nguyên tố.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi 2 số a, b (1<a<b<106)

**Output**

* Với mỗi bộ test, ghi ra số lượng số thỏa mãn trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  10 100  1234 5678 | 4  26 |

#include<stdio.h>

#include<math.h>

int nt(int a){

if(a<2) return 0;

else{

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

if(a%i==0) return 0;

}

return 1;

}

}

int ktsnt(int a){

if(nt(a)==0) return 0;

else{

while(a>0){

int d=a%10;

if(nt(d)==0) return 0;

a=a/10;

}

return 1;

}

}

int main(){

int t;

scanf("%d",&t);

while(t--){

int a,b;

int d=0;

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

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

if(ktsnt(i)==1) d++;

}

printf("%d",d);

printf("\n");

}

}

PHÂN TÍCH THỪA SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03033/top)

Hãy phân tích một số nguyên dương thành tích các thừa số nguyên tố.

**Input**

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số nguyên dương n không quá 9 chữ số.

**Output**

Mỗi bộ test viết ra kết quả phân tích theo mẫu như trong ví dụ (chú ý: giữa các số với dấu \* có một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  60  128  10000 | 60 = 2^2 \* 3^1 \* 5^1  128 = 2^7  10000 = 2^4 \* 5^4 |

#include<stdio.h>

#include<math.h>

void nt(long a){

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

int d=0;

while(a%i==0){

d++;

if(i==a) printf(" %d^%d\n",i,d);

a=a/i;

if(a%i!=0 && a!=1) printf(" %d^%d \*",i,d);

}

}

if(a>1) printf(" %d^1\n",a);

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long a;

scanf("%ld",&a);

printf("%ld =",a);

nt(a);

}

}

SỐ SMITH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03040/top)

Một số Smith là một số tự nhiên thỏa mãn tổng các chữ số của nó bằng với tổng các chữ số của các thừa số nguyên tố của nó.

Nhiệm vụ của bạn là hãy xác định xem số nguyên N đã cho có là số Smith hay không?

**Ví dụ 1:**

378 = 2 x 3 x 3 x 3 x 7

Tổng các chữ số là 3 + 7 + 8 = 18.

Tổng các chữ số của các nhân tử là 2 + 3 + 3 + 3 + 7 = 18.

**Ví dụ 2:**

4937775 = 3 x 5 x 5 x 65837.

Ta có 4+9+3+7+7+7+5 = 42 = 3+5+5+6+5+8+3+7.

**Input**

* Chỉ có một số nguyên dương N không quá 9 chữ số.

**Output**

* In ra “YES” nếu số đã cho là số Smith, in ra “NO” trong trường hợp ngược lại.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 378 | YES |
| 120 | NO |

#include <stdio.h>

#include<stdio.h>

#include<math.h>

long tong1(long n){

long s=0;

while(n>0){

s=s+n%10;

n=n/10;

}

return s;

}

long tong2(long n){

long s=0;

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

while(n%i==0){

int j=i;

while(j>0){

s=s+j%10;

j=j/10;

}

n=n/i;

}

}

if(n>1){

while(n>0){

s=s+n%10;

n=n/10;

}

}

return s;

}

int main(){

long a;

scanf("%ld",&a);

if(tong1(a)!=tong2(a) || a<2) printf("NO\n");

else printf("YES\n");

}

HÌNH VUÔNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03041/top)

Cho 2 đỉnh là góc dưới bên trái và góc trên bên phải của hình chữ nhật trong hệ tọa độ Oxy. Hãy kiểm tra xem đó có phải là hình vuông hay không.

**Input**

Dòng đầu ghi số bộ test

Mỗi test có 4 số nguyên a,b,c,d.

Trong đó (a,b) là tọa độ điểm góc dưới bên trái, (c,d) là tọa độ góc trên bên phải (-1000 <a<c<1000; -1000<b<d<1000)

(a luôn nhỏ hơn c; b luôn nhỏ hơn d).

**Output**

Ghi ra YES hoặc NO tương ứng với kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1 1 3 3  1 2 3 7 | YES  NO |

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

int x1,y1,x2,y2;

scanf("%d %d %d %d",&x1,&y1,&x2,&y2);

if(x2-x1==y2-y1) printf("YES\n");

else printf("NO\n");

}

}

SỐ GIẢM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03042/top)

Một số gọi là **số giảm** nếu các chữ số từ trái qua phải tạo thành dãy giảm (tức là chữ số phía sau nhỏ hơn chữ số ngay trước). Viết chương trình đếm các số giảm có trong đoạn [a,b].

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi 2 số a,b (1<a<b<108)

**Output**

* Với mỗi bộ test, ghi ra số lượng các số giảm đếm được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  10 100 | 45 |

#include<stdio.h>

int kt(long long a){

int e=a%10;

a=a/10;

int d=a%10;

while(a>0){

if(e>=d) return 0;

e=d;

a=a/10;

d=a%10;

}

return 1;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long a,b,dem=0;

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

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

if(kt(i)==1) dem++;

}

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

}

}

CHỮ SỐ 4

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03046/top)

Nam rất ghét chữ số 4 nên không muốn xuất hiện chữ số 4 trong các số may mắn của mình. Dãy số may mắn của Nam có N chữ số và phải thỏa mãn tính chất thuận nghịch, tổng chữ số phải chia hết cho 10.

Cho số N, hãy liệt kê các số may mắn của Nam với đúng N chữ số.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi bộ test ghi số N (1<N<7)

**Output**

* Với mỗi bộ test, ghi ra lần lượt các số may mắn có N chữ số, các số cách nhau một khoảng trống.
* Hết một bộ test thì xuống dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 55  181 262 505 686 767 929 |

#include<stdio.h>

int kt(long a){

long s=0,m=a,b=0;

while(a>0){

int d=a%10;

if(d==4) return 0;

s=s+d;

b=b\*10+d;

a=a/10;

}

if(s%10==0 && b==m) return 1;

return 0;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

int n;

scanf("%d",&n);

long a,b=1;

while(n--){

b\*=10;

}

a=b/10;

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

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

}

printf("\n");

}

}

CÂN BẰNG CHẴN LẺ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03047/top)

Một số được gọi là “cân bằng chẵn lẻ” nếu số chữ số chẵn và số chữ số lẻ là bằng nhau. Tất nhiên khi đó số chữ số của nó phải là chẵn.

Cho số N là một số chẵn (1<N<7). Hãy liệt kê các số cân bằng chẵn lẻ có N chữ số. Mỗi dòng ghi ra 10 số thỏa mãn.

**Input**

Chỉ có duy nhất số N (chẵn)

**Output**

Ghi ra các số cân bằng chẵn lẻ có N chữ số theo thứ tự từ nhỏ đến lớn. Mỗi dòng ghi 10 số.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | 10 12 14 16 18 21 23 25 27 29  30 32 34 36 38 41 43 45 47 49  50 52 54 56 58 61 63 65 67 69  70 72 74 76 78 81 83 85 87 89  90 92 94 96 98 |

#include<stdio.h>

#include<math.h>

int kt(long a){

long l=0,c=0;

while(a>0){

int d=a%10;

if(d%2==1) l++;

if(d%2==0) c++;

a=a/10;

}

if(l==c) return 1;

return 0;

}

int main(){

int n;

scanf("%d",&n);

long a,b=1;

while(n--){

b=b\*10;

}

a=b/10;

int dem=0;

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

if(kt(i)==1){

dem++;

printf("%ld ",i);

}

if(dem==10){

printf("\n");

dem=0;

}

}

}

SỐ ƯU THẾ CHẴN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03048/top)

Một số được gọi là số ưu thế chẵn nếu nó là số chẵn và số chữ số chẵn nhiều hơn số chữ số lẻ. Hãy viết chương trình kiểm tra một số có phải ưu thế chẵn hay không.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi dòng tiếp theo ghi một số nguyên dương không quá 18 chữ số

**Output**

* Ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  12345  22566678800 | NO  YES |

#include<stdio.h>

#include<math.h>

int kt(long long a){

long long l=0,c=0;

if(a%2==1) return 0;

while(a>0){

int d=a%10;

if(d%2==1) l++;

if(d%2==0) c++;

a=a/10;

}

if(l<c) return 1;

return 0;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long long n;

scanf("%lld",&n);

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

else printf("NO\n");

}

}

SỐ ƯU THẾ LẺ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03049/top)

Một số được gọi là số ưu thế lẽ nếu nó là số lẻ và số chữ số lẻ nhiều hơn số chữ số chẵn. Hãy viết chương trình kiểm tra một số có phải ưu thế lẻ hay không.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi dòng tiếp theo ghi một số nguyên dương không quá 18 chữ số

**Output**

* Ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  12345  22566678801 | YES  NO |

#include<stdio.h>

#include<math.h>

int kt(long long a){

long long l=0,c=0;

if(a%2==0) return 0;

while(a>0){

int d=a%10;

if(d%2==1) l++;

if(d%2==0) c++;

a=a/10;

}

if(l>c) return 1;

return 0;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

long long n;

scanf("%lld",&n);

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

else printf("NO\n");

}

}

KHOẢNG CÁCH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01046/top)

Cho 2 điểm A trong không gian Oxy. Hãy tính khoảng cách giữa hai điểm đó.

**Input**

* Dòng đầu ghi số bộ test, không quá 20.
* Mỗi bộ test có 4 số thực lần lượt là tọa độ của 2 điểm A và B, giá trị tuyệt đối không quá 1000.

**Ouput**

Với mỗi bộ test, viết ra khoảng cách giữa 2 điểm với 4 chữ số sau dấu phẩy.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  0 0 0 5  0 199 5 6 | 5.0000  193.0648 |

#include<stdio.h>

#include<math.h>

main(){

int t;

scanf("%d",&t);

while(t--){

float x1,y1,x2,y2;

scanf("%f %f %f %f",&x1,&y1,&x2,&y2);

/\* float kc=(float) sqrt((x1-x2)\*(x1-x2) + (y1-y2)\*(y1-y2));\*/

printf("%.4f\n",sqrt((x1-x2)\*(x1-x2) + (y1-y2)\*(y1-y2)));

}

}

ĐIỂM TRUNG TÂM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03050/top)

Trên tờ giấy A4, Nam đã đánh dấu sẵn N điểm từ 1 đến N và liệt kê danh sách N-1 đoạn thẳng nối các điểm với nhau. Nam nhờ Bình kiểm tra giúp xem với danh sách N-1 đoạn thẳng đó thì có điểm nào là điểm trung tâm hay không.

*Một điểm được coi là trung tâm nếu nó có đoạn thẳng nối với tất cả N-1 điểm còn lại.*

**Input**

* Dòng đầu tiên ghi số N (1 ≤ N ≤ 105).
* N-1 dòng tiếp theo, mỗi dòng ghi ra một đoạn thẳng.

**Kết quả**

* Ghi ra trên một dòng chữ **“Yes”** nếu tồn tại điểm trung tâm; chữ **“No”** trong trường hợp ngược lại.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 1 2 1 3 1 4 1 5 | Yes |

#include<stdio.h>

int c[1000000];

int main(){

int n,a,b;

scanf("%d",&n);

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

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

c[a]++;

c[b]++;

}int kt=0;

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

if(c[i]>=n-1){

kt=1;break;

}

}

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

else printf("Yes");

}

CẶP SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03053/top)

Cho số nguyên dương chẵn N>2. Hãy liệt kê các cặp số nguyên tố p, q có tổng đúng bằng N. Ví dụ N = 6 ta có 1 cặp số nguyên tố là 3 + 3 =6.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số chẵn N.
* T, N thỏa mãn ràng buộc : 1≤T≤100; 4≤N≤10000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 6 32 | 3 3 3 29 13 19 |

#include<stdio.h>

#include<math.h>

int b[10099];

main(){

int t;

scanf("%d",&t);

for(int i=2;i<10009;i++){

if(b[i]==0){

for(int j=i\*2;j<10009;j=j+i){

b[j]=1;

}

}

}

while(t--){

int n;

scanf("%d",&n);

int k=0;

int a[10001];

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

if(b[i]==0){

a[k++]=i;

}

}

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

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

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

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

}

}

}

printf("\n");

}

}

Giá trị trung bình của một mảng

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/LAB01-0007/top)

Cho một dãy N các phần tử nguyên. Tính giá trị trung bình (để sau dấu phẩy đúng 3 chữ số)

Input:

Hai dòng, dòng đầu là số phần tử; dòng tiếp theo sau là dòng các phần tử

Output:

Giá trị trung bình của dãy với 3 chữ số sau dấu phảy

| Ví dụ minh họa | |
| --- | --- |
| **Input** | **Output** |
| 4  2  4   1  2 | 2.250 |

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

int i;

int a[n];

long s=0;

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

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

s=s+a[i];

}

float d=(float) s/n;

printf("%.3f",d);

}

 CẮT ĐÔI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03054/top)

Với một vài số nguyên dương có 1 chữ số, khi cắt đôi số đó theo chiều ngang và lấy nửa phía trên thì ta vẫn có một số nguyên. Cụ thể:

* Số 0 cắt đôi vẫn ra số 0
* Số 1 cắt đôi vẫn ra số 1
* Số 8 cắt đôi ra số 0
* Số 9 cắt đôi ra số 0
* Các số khác cắt đôi sẽ không hợp lệ.

Cho một số nguyên dương không quá 18 chữ số. Hãy in ra kết quả “cắt đôi” của số đó.

Nếu không hợp lệ thì ghi ra INVALID. Chú ý: nếu cắt đôi ra một dãy toàn 0 thì cũng được coi là không hợp lệ. Kết quả cắt đôi thì không tính chữ số 0 ở đầu.

**Input**

Dòng đầu ghi số bộ test. Mỗi bộ test ghi một số nguyên dương không quá 18 chữ số.

**Output**

Ghi ra kết quả tính toán

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1890  3681  8919 | 1000  INVALID  10 |

#include<stdio.h>

main(){

int t;

scanf("%d",&t);

while(t--){

long long n;

scanf("%lld",&n);

long long x=1,kt=0,e,s=0;

while(n>0){

int d=n%10;

if(d!=1 && d!=0 && d!=8 && d!=9){

kt=1;

break;

}

if(d==1) e=1;

if(d==0 || d==8 || d==9) e=0;

s=e\*x+s;

x=x\*10;

n=n/10;

}

if(kt==1 || s==0) printf("INVALID\n");

else printf("%lld\n",s);

}

}

MÃ SỐ QUỐC GIA

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03055/top)

Trong mã hàng hóa  người ta thường ghi kèm theo mã số quốc gia sản xuất. Nếu sản xuất tại Việt Nam thì mã tương ứng là 084. Bài toán đặt ra là cho một dãy mã dạng số nguyên không quá 18 chữ số. Hãy loại bỏ đoạn mã 084 ra khỏi mã ban đầu.

Dữ liệu đảm bảo dãy mã luôn có duy nhất một lần cụm 084.

**Input**

Dòng đầu ghi số bộ test. Mỗi test là một số nguyên có ít nhất 4 chữ số nhưng không quá 18 chữ số.

**Output**

Ghi ra kết quả sau khi loại bỏ 084

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  123084567  3300478808445  1084 | 123567  3300478845  1 |

#include<stdio.h>

#include<string.h>

int b[1000];

main(){

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

char a[20];

gets(a);

int n=strlen(a);

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

if(a[i]=='0' && a[i+1]=='8'&& a[i+2]=='4'){

b[i]=1;

b[i+1]=1;

b[i+2]=1;

}

}

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

if(b[i]!=1) printf("%c",a[i]);

}

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

b[i]=0;

}

printf("\n");

}

}

 SỐ ĐẸP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03061/top)

Một số được coi là đẹp nếu chữ số đầu gấp đôi chữ số cuối hoặc ngược lại; đồng thời các chữ số  từ vị trí thứ 2 đến gần cuối thỏa mãn là một số thuận nghịch.

Ví dụ: các số 36788766; 12345654322 là các số đẹp.

Viết chương trình kiểm tra số đẹp theo tiêu chí trên.

**Input**

* Dòng đầu ghi số bộ test
* Mỗi test là một số nguyên dương không quá 18 chữ số

**Output**

* Ghi ra YES tương ứng với số đẹp, NO trong trường hợp ngược lại

Ví dụ

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  36788766  22345654321  12345654321 | YES  YES  NO |

#include<stdio.h>

#include<string.h>

int b[1000];

int kt(char a[]){

int l=strlen(a);

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

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

}

if(a[0]!=2\*a[l-1] && a[0]\*2!=a[l-1]) return 0;

for(int i=1;i<l/2;i++){

if(a[i]!=a[l-1-i]) return 0;

}

return 1;

}

main(){

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

char a[20];

gets(a);

if(kt(a)==1) printf("YES");

else printf("NO");

printf("\n");

}

}

MA TRẬN XOẮN ỐC 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05016/top)

Ma trận xoắn ốc cấp N là một ma trận vuông cấp N\*N trong đó ghi các số nguyên dương tăng dần từ 1 đến N\*N được điền theo thứ tự xoắn ốc từ ngoài vào trong.

Hãy viết chương trình in ra ma trận xoắn ốc cấp N.

**Input:**

Chỉ có một dòng ghi số N (1 ≤ N ≤ 100)

**Ouput:**

Ghi ra ma trận kết quả có N dòng, mỗi giá trị số cách nhau một khoảng trống.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1 2 3 4 5  16 17 18 19 6  15 24 25 20 7  14 23 22 21 8  13 12 11 10 9 |

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

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

int x=1,k=0,i,j;

int a[n][n];

while(x<=p){

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

a[k][j]=x++;

}

for(i=k+1;i<m;i++){

a[i][m-1]=x++;

}

for(j=m-2;j>=k;j--){

a[m-1][j]=x++;

}

for(i=m-2;i>k;i--){

a[i][k]=x++;

}

m--;

k++;

}

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

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

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

}

printf("\n");

}

}

TẬP TỪ RIÊNG CỦA HAI XÂU 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06017/top)

Cho hai chuỗi ký tự S1 và S2 độ dài không quá 100.

Hãy viết chương trình tìm các từ chỉ xuất hiện trong S1 mà không xuất hiện trong S2. Chú ý: mỗi từ chỉ liệt kê 1 lần và sắp xếp theo thứ tự xuất hiện trong từ điển  
**Input:**

Có hai dòng, dòng thứ nhất ghi xâu S1, dòng thứ 2 ghi xâu S2.

**Output:**

Ghi lần lượt các từ thỏa mãn trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| abc ab ab ab abcd  ab abc | abcd |

#include<stdio.h>

#include<string.h>

int main(){

char a[101],b[101],s1[101][101],s2[101][101], t[101];

gets(a);

gets(b);

int la=strlen(a), lb=strlen(b), i, j=0,k=0,n=0,m=0;

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

if(a[i]!=' ') s1[n][j++]=a[i];

if(a[i]==' ' || i==la-1) {

n++;

j=0;

}

}

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

if(b[i]!=' ') s2[m][k++]=b[i];

if(b[i]==' ' || i==lb-1) {

m++;

k=0;

}

}

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

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

if(strcmp(s1[i],s1[j])>0){

strcpy(t,s1[j]);

strcpy(s1[j],s1[i]);

strcpy(s1[i],t);

}

}

}

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

int kt=0;

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

if(strcmp(s1[i],s1[j])==0){

kt=1;

break;

}

}

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

if(strcmp(s1[i],s2[j])==0){

kt=1;

break;

}

}

if(kt==0) printf("%s ",s1[i]);

}

}

SỐ ĐẸP 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03020/top)

Một số được coi là số đẹp nếu nó là số thuận nghịch, có chứa ít nhất một chữ số 6, và tổng các chữ số của nó có chữ số cuối cùng là 8.

Viết chương trình liệt kê trong một đoạn giữa hai số nguyên [a,b] cho trước có bao nhiêu số đẹp như vậy

**Input:**

Chỉ có một dòng ghi hai số a,b.

**Output:**

Ghi ra kết quả tính toán trên một dòng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 500 | 161 |

#include<stdio.h>

int main(){

int m,n;

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

if(m>n){

int z=m;

m=n;

n=z;

}

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

int x=i,s=0,k=0,kt=0;

while(x>0){

int d=x%10;

if(d==6){

kt++;

}

s=s\*10+d;

k=k+d;

x=x/10;

}

if(kt>0 && k%10==8 && s==i) printf("%d ",i);

}

}

 TAM GIÁC PASCAL

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03028/top)

Tam giác Pascal là tam giác có công thức tính giá trị một vị trí bất kỳ như sau  
(n,k) = n!/k!(n-k)!  
Trong đó: n là hàng và k là cột.

Người ta thường tận dụng hàm đệ quy để điền các giá trị cho tam giác Pascal.

Viết chương trình sử dụng đệ quy nhập vào số nguyên dương n không quá 10 và in ra tam giác Pascal tương ứng (lưu ý mỗi phần tử cách nhau 1 khoảng trắng)

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1  1 1  1 2 1  1 3 3 1  1 4 6 4 1 |

#include<stdio.h>

int gt(int a){

int i,s=1;

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

s=s\*i;

}

return s;

}

int main(){

int n,i,j,x;

scanf("%d",&n);

int a[10][10];

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

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

if(j==0 || j==i) {

x=1;

}

else

{

x=gt(i)/(gt(j)\*gt(i-j));

}

printf("%d ",x);

}

printf("\n");

}

}

HÌNH VUÔNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02014/top)

Nhập vào kích thước hình vuông (không quá 9). Vẽ hình vuông số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi kích thước hình vuông (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 4444444  4333334  4322234  4321234  4322234  4333334  4444444 |

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

int m=2\*n-1;

int x=n,k=0,i,j;

int a[2\*n-1][2\*n-1];

while(x>0){

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

a[k][j]=x;

}

for(i=k+1;i<m;i++){

a[i][m-1]=x;

}

for(j=m-2;j>=k;j--){

a[m-1][j]=x;

}

for(i=m-2;i>k;i--){

a[i][k]=x;

}

m--;

k++;

x--;

}

for(i=0;i<2\*n-1;i++){

for(j=0;j<2\*n-1;j++){

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

}

printf("\n");

}

}

TỔNG CHỮ SỐ CHIA HẾT CHO 5

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03022/top)

Viết chương trình nhập vào số nguyên dương n không quá 5 chữ số. Thực hiện liệt kê các số nguyên tố nhỏ hơn n và thỏa mãn có tổng các chữ số chia hết cho 5. Có bao nhiêu số như vậy.

**Input**

Chỉ có một dòng ghi số n.

**Output**

Gồm hai dòng: dòng đầu liệt kê các số thỏa mãn, dòng thứ 2 ghi số lượng các số tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 100 | 5 19 23 37 41 73  6 |

#include<stdio.h>

#include<math.h>

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int tcn(int a){

int s=0;

while(a>0){

s=s+a%10;

a=a/10;

}

return s;

}

int main(){

int n,d=0;

scanf("%d",&n);

for(int i=5;i<n;i++){

if(nt(i)==1 && tcn(i)%5==0){

d++;

printf("%d ",i);}

}

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

}

SỐ THUẬN NGHỊCH KHÔNG CÓ CHỮ SỐ 9

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03023/top)

Viết chương trình nhập vào số N không quá 4 chữ số, thực hiện liệt kê các số thuận nghịch lớn hơn 1 và nhỏ hơn N thỏa mãn không chứa chữ số 9. Có bao nhiêu số như vậy.

**Input**

Chỉ có một dòng ghi số n.

**Output**

Gồm hai dòng: dòng đầu liệt kê các số thỏa mãn, dòng thứ 2 ghi số lượng các số tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 100 | 2 3 4 5 6 7 8 11 22 33 44 55 66 77 88 15 |

#include<stdio.h>

int sothuannghich(int a){

int s=0,d;

int x=a;

while(a>0){

if(a%10==9) {

return 0;

break;

}

else s=s\*10+a%10;

a=a/10;

}

if(s==x) return 1;

else return 0;

}

int main(){

int n,d=0;

scanf("%d",&n);

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

if(sothuannghich(i)==1) {

printf("%d ",i);

d++;

}

}

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

}

TAM GIÁC SỐ - 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02015/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1  123  12345  1234567  123456789 |

#include<stdio.h>

int main(){

int n,d=1;

scanf("%d",&n);

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

for(int j=1;j<=2\*d-1;j++){

printf("%d",j);

}

d++;

printf("\n");

}

}

TAM GIÁC SỐ - 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02016/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1  24  135  2468  13579 |

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

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

if(i%2==0){

for(int j=1;j<=2\*i+1;j++){

if(j%2==1) printf("%d",j);

}

}

else {

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

if(j%2==0) printf("%d",j);

}

}

printf("\n");

}

}

TAM GIÁC SỐ - 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02017/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 1  131  13531  1357531  135797531 |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

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

int d=-1;

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

d=d+2;

printf("%d",d);

}

for(j=i+2;j<=2\*i+1;j++){

d=d-2;

printf("%d",d);

}

printf("\n");

}

}

TAM GIÁC SỐ - 4

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02018/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | ~~~~1  ~~~131  ~~13531  ~1357531  135797531 |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

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

for(j=n-1-i;j>=1;j--){

printf("~");

}

int d=-1;

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

d=d+2;

printf("%d",d);

}

for(j=i+2;j<=2\*i+1;j++){

d=d-2;

printf("%d",d);

}

printf("\n");

}

}

TAM GIÁC SỐ - 5

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02019/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 2  242  24642  2468642  2468108642 |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

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

int d=0;

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

d=d+2;

printf("%d",d);

}

for(j=i+2;j<=2\*i+1;j++){

d=d-2;

printf("%d",d);

}

printf("\n");

}

}

TAM GIÁC SỐ - 6

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02020/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | ~~~~2  ~~~242  ~~24642  ~2468642  2468108642 |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

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

for(j=n-1-i;j>=1;j--){

printf("~");

}

int d=0;

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

d=d+2;

printf("%d",d);

}

for(j=i+2;j<=2\*i+1;j++){

d=d-2;

printf("%d",d);

}

printf("\n");

}

}

TAM GIÁC SỐ - 7

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02021/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

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

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);int m=1;

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

int e=n-1,d=m;

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

printf("%d ",d);

d=d+e;

e--;

}

m++;

printf("\n");

}

}

TAM GIÁC SỐ - 8

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02022/top)

Nhập vào chiều cao tam giác (không quá 9).

Vẽ hình tam giác số theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 9).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 1  3 2  4 5 6  10 9 8 7 |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

int d=0,m,a,x;

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

if(i%2==1){

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

d++;

printf("%d ",d);

}

x=d;

}

else {

m=x+i;

a=m;

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

printf("%d ",m);

m--;

}

}

d=a;

printf("\n");

}

}

 SẮP XẾP CHẴN LẺ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04025/top)

Viết chương trình nhập vào mảng A gồm n phần tử (1 < n < 100). Thực hiện sắp xếp tăng dần các phần tử chẵn và lẻ của mảng và In ra.  
Trong đó:  
INPUT  
Hàng thứ nhất là số phần tử n của mảng A  
Hàng thứ hai là các phần tử của mảng A  
OUTPUT  
Mảng kết quả in lần lượt dãy số chẵn sau đó đến dãy số lẻ.

**Ví dụ:**  
INPUT  
8  
1 3 8 2 9 7 6 5  
OUTPUT  
2 6 8 1 3 5 7 9

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

long long a[100];

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

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

}

long long tg;

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

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

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

tg = a[i];

a[i] = a[j];

a[j] = tg;

}

}

}

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

if(a[i]%2==0) printf("%lld ",a[i]);

}

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

if(a[i]%2!=0) printf("%lld ",a[i]);

}

}

TÌM HÀNG NHIỀU SỐ NGUYÊN TỐ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05026/top)

Viết chương trình nhập vào ma trận vuông cấp n (1 < n < 30) chỉ bao gồm các số nguyên dương không quá 1000.

Thực hiện tìm hàng có nhiều phần tử là số nguyên tố nhất. Nếu có nhiều hơn một hàng thỏa mãn thì in ra hàng đầu tiên.

INPUT  
- Dòng đầu tiên là cấp của ma trận  
- Các dòng tiếp theo là các phần tử của ma trận  
OUTPUT  
- Dòng đầu tiên là số thứ tự hàng thỏa mãn  
- Dòng thứ hai liệt kê các phần tử thỏa mãn

**Ví dụ:**  
INPUT  
4  
8 2 3 4  
5 6 7 8  
9 10 11 12  
13 14 15 16  
OUTPUT  
1  
2 3

#include<stdio.h>

#include<math.h>

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

int n;

scanf("%d",&n);

int a[n][n];

int i,j,d,max=0,vt;

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

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

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

}

scanf("\n");

}

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

d=0;

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

if(nt(a[i][j])==1){

d++;

}

}

if(d>max){

max=d;

vt=i;

}

}

printf("%d\n",vt+1);

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

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

if(i==vt && nt(a[i][j])==1) printf("%d ",a[i][j]);

}

}

}

TỔNG CÁC SỐ NGUYÊN TỐ THUỘC TAM GIÁC TRÊN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05010/top)

Viết chương trình nhập ma trận A là ma trận vuông cấp n (không quá 50). Tìm tổng các phần tử là số nguyên tố thuộc tam giác trên.  
  
INPUT  
- Dòng đầu tiên là cấp của ma trận  
- Các dòng tiếp theo là các phần tử của ma trận  
OUTPUT  
- Ghi ra kết quả của chương trình

Ví dụ:

INPUT  
4  
1 2 3 4  
5 6 7 8  
9 10 11 12  
13 14 15 16  
OUTPUT  
23

#include<stdio.h>

#include<math.h>

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

int n,i,j,s=0;

scanf("%d",&n);

int a[n][n];

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

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

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

}

}

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

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

if(nt(a[i][j])==1) {

s+=a[i][j];

}

}

}

printf("%d",s);

}

MA TRẬN XOẮN ỐC FIBONACCI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05020/top)

Ma trận xoáy ốc Fibonacci cấp N là ma trận vuông có N\*N phần tử. Các số được điền vào ma trận theo chiều kim đồng hồ đều là các số thuộc dãy Fibonacci từ nhỏ đến lớn (tính từ số 0).

Viết chương trình nhập N (không quá 9) và in ra ma trận xoắn ốc Fibonacci tương ứng.

**Ví dụ:**

INPUT  
3  
OUTPUT  
0 1 1  
13 21 2  
8 5 3

#include<stdio.h>

int fb(int a){

if(a==0 ) return 0;

else if (a==1||a==2) return 1;

else{

int x=1;

int y=1;

int z=0;

int d=3;

while(d<=a){

z=x+y;

x=y;

y=z;

d++;

}

return z;

}

}

int main(){

int n;

scanf("%d",&n);

int p=n\*n;

int m=n;

int k=0,i,j;

int a[n][n];

int d=0;

while(d<p){

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

a[k][j]=fb(d);

d++;

}

for(i=k+1;i<m;i++){

a[i][m-1]=fb(d);

d++;

}

for(j=m-2;j>=k;j--){

a[m-1][j]=fb(d);

d++;

}

for(i=m-2;i>k;i--){

a[i][k]=fb(d);

d++;

}

m--;

k++;

}

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

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

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

}

printf("\n");

}

}

PHÂN TÍCH THỪA SỐ NGUYÊN TỐ 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01031/top)

Viết chương trình phân tích một số nguyên dương (không quá 6 chữ số) thành tích các thừa số nguyên tố.

Kết quả được viết theo mẫu trong Ví dụ (có chữ x giữa các thừa số)

**Input:**

Chỉ có một dòng ghi số n.

**Output:**

Ghi ra kết quả tính toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 28 | 2x2x7 |

#include<stdio.h>

int main(){

int n,i;

scanf("%d",&n);

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

if(n%i==0) printf("%d",i);

n=n/i;

break;

}

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

while(n%i==0){

printf("x%d",i);

n=n/i;

}

}

}

TAM GIÁC KÝ TỰ - 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02027/top)

Nhập vào chiều cao tam giác (không quá 20).

Vẽ hình tam giác ký tự theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 20).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | a  c b  d e f  j i h g |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

int d=0,m,a,x;

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

if(i%2==1){

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

d++;

printf("%c ",d+96);

}

x=d;

}

else {

m=x+i;

a=m;

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

printf("%c ",m+96);

m--;

}

}

d=a;

printf("\n");

}

}

TAM GIÁC KÝ TỰ - 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02029/top)

Nhập vào chiều cao tam giác (không quá 20).

Vẽ hình tam giác ký tự theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 20).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | A  B E  C F H  D G I J |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);int m=1;

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

int e=n-1,d=m;

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

printf("%c ",d+64);

d=d+e;

e--;

}

m++;

printf("\n");

}

}

TAM GIÁC KÝ TỰ - 4

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C02030/top)

Nhập vào chiều cao tam giác (không quá 20).

Vẽ hình tam giác ký tự theo nguyên tắc tương ứng theo mẫu trong ví  dụ dưới đây.

**Input**

Chỉ có một số dòng ghi chiều cao (không quá 20).

**Output**

Ghi ra kết quả theo mẫu.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | @  @B@  @BDB@  @BDFDB@  @BDFHFDB@ |

#include<stdio.h>

int main(){

int n,i,j;

scanf("%d",&n);

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

int d=0;

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

d=d+2;

printf("%c",d+62);

}

for(j=i+2;j<=2\*i+1;j++){

d=d-2;

printf("%c",d+62);

}

printf("\n");

}

}

CHUYỄN ĐỔI HAI HÀNG TRONG MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05006/top)

Viết chương trình  nhập vào ma trận nguyên dương cấp M\*N (không quá 50). Chuyển đổi hai hàng a và b trong ma trận và in ra kết quả.  
  
INPUT  
- Hàng thứ nhất là cấp m,n của ma trận  
- m hàng tiếp theo là các phần tử của ma trận  
- Hàng cuối cùng là giá trị a,b. Là các hàng cần chuyển đổi  
OUTPUT  
- Ma trận kết quả

**Ví dụ:**

INPUT  
3 3  
1 2 3  
4 5 6  
7 8 9  
1 3

OUTPUT  
7 8 9  
4 5 6  
1 2 3

#include<stdio.h>

int main(){

int n,m,i,j;

int x,y;

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

int a[m][n];

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

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

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

}

}

scanf("%d %d",&x,&y);

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

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

if(i==x-1) printf("%d ",a[y-1][j]);

else if(i==y-1) printf("%d ",a[x-1][j]);

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

}

printf("\n");

}

}

CHUYỂN ĐỔI HAI CỘT TRONG MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05007/top)

Viết chương trìnhnhập vào ma trận nguyên dương cấp M\*N (không quá 50). Chuyển đổi hai cột a và b trong ma trận và in ra kết quả.

INPUT  
- Hàng thứ nhất là cấp m,n của ma trận  
- m hàng tiếp theo là các phần tử của ma trận  
- Hàng cuối cùng là vị trí cột cần chuyển đổi  
OUTPUT  
- Ma trận kết quả

**Ví dụ:**

INPUT  
3 3  
1 2 3  
4 5 6  
7 8 9  
1 3  
OUTPUT  
3 2 1  
6 5 4  
9 8 7

#include<stdio.h>

int main(){

int n,m,i,j;

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

int a[m][n];

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

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

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

}

}

int x,y;

scanf("%d %d",&x,&y);

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

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

if(j==x-1) printf("%d ",a[i][y-1]);

else if(j==y-1) printf("%d ",a[i][x-1]);

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

}

printf("\n");

}

}

CHUYỂN ĐỔI HAI ĐƯỜNG CHÉO

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05009/top)

Viết chương trình nhập vào ma trận vuông các số nguyên dương cấp M (không quá 50). Thực hiện chuyển đổi hai đường chéo của ma trận và in ra ma trận kết quả.  
  
INPUT  
- Hàng thứ nhất là cấp m của ma trận  
- m hàng tiếp theo là các phần tử của ma trận  
OUTPUT  
- Ma trận kết quả

**Ví dụ:**

INPUT  
3  
1 2 3  
4 5 6  
7 8 9  
OUTPUT  
3 2 1  
4 5 6  
9 8 7

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

int i,j,a[n][n];

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

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

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

}

}

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

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

if(i==j) printf("%d ",a[i][n-1-j]);

else if(j==n-1-i) printf("%d ",a[i][i]);

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

}

printf("\n");

}

}

ĐẾM SỐ LẦN XUẤT HIỆN CÁC TỪ TRONG XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06005/top)

Tìm số lần xuất hiện các từ trong xâu S (không phân biệt chữ hoa chữ thường). Kết quả in ra từ (ở dạng in thường) và số lần xuất hiện của chúng

**Input:**

Chỉ có một dòng ghi xâu S.

**Output:**

Ghi ra kết quả đếm từng từ theo thứ tự xuất hiện.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| aaa bbb ccc AA bb aa ccc aa | aaa 1  bbb 1  ccc 2  aa 3  bb 1 |

#include<stdio.h>

#include<string.h>

int main(){

char a[10001];

gets(a);

int l=strlen(a);

int i,j,x,y,k,e;

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

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

}

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

if(a[i]!=' ' && a[i]!='\0' && (i==0 || a[i-1]==' ' || a[i-1]=='\0')){

x=i;

e=1;

for(j=x;j<=l;j++){

y=j;

if(a[j]==' '|| a[j]=='\0') break;

}

k=x;

for(j=y+1;j<l;j++){

if(a[j]!=' ' && a[j]!='\0' && (a[j-1]==' ' || a[j-1]=='\0')){

int z=j;

while(a[j]==a[k] && (x==0 || a[x-1]==' ' || a[x-1]=='\0') && k<y){

j++;

k++;

}

if((a[j]==' ' || a[j]=='\0') && (a[k]==' ' || a[k]=='\0')){

e++;

for(int d=z;d<j+1;d++){

a[d]='\0';

}

}

k=x;

}

}

for(j=x;j<y;j++){

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

if(j==y-1) printf(" %d\n",e);

}

}

}

}

CHÈN XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06016/top)

Cho xâu S1 và xâu S2, độ dài không quá 100.

Hãy chèn xâu S2 vào vị trí p trong xâu S1 (vị trí ký tự đầu tiên là vị trí 1).

**Input:**

Dòng thứ nhất ghi xâu S1

Dòng thứ hai ghi xâu S2

Dòng thứ ba ghi số p (giá trị p đảm bảo nằm trong phạm vi xâu S1)

**Output:**

Ghi ra kết quả.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| mon thcs2 hoc de  ngon ngu C.  1 | ngon ngu C.mon thcs2 hoc de |

#include<stdio.h>

#include<string.h>

int main()

{

char str1[255], str2[255];

int k;

gets(str1);

gets(str2);

scanf("%d",&k);

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

{

printf("%c",str1[i]);

}

for(int i=0;i<strlen(str2);i++)

{

printf("%c",str2[i]);

}

for(int i=k-1;i<strlen(str1);i++)

{

printf("%c",str1[i]);

}

}

ĐỊA CHỈ EMAIL PTIT - 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06019/top)

Địa chỉ email của cán bộ PTIT được cấp theo nguyên tắc ghép tên với chữ cái đầu tiên của họ và tên đệm. Viết chương trình cho phép tạo các địa chỉ email theo tên cán bộ(có thể không chuẩn).

**Input:**

Chỉ có một dòng ghi xâu họ tên (độ dài không quá 50)

**Output:**

Ghi ra kết quả.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ngUYEN van nam | nvnam@ptit.edu.vn |

#include<stdio.h>

#include<string.h>

int main(){

char a[51];

gets(a);

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

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

d++;

}

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

l--;

}

if(a[d]>='A' && a[d]<='Z'){

a[d]=a[d]+32;

}

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

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

if(a[i]==' ' && a[i+1]!=' ') {

if(a[i+1]>='A' && a[i+1]<='Z'){

a[i+1]=a[i+1]+32;

}

printf("%c",a[i+1]);

vt=i;

}

}

for(i=vt+2;i<l;i++){

if(a[i]>='A' && a[i]<='Z'){

a[i]=a[i]+32;

}

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

}

printf("@ptit.edu.vn");

}

ĐẾM CHỮ SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01065/top)

Viết chương trình nhập vào một số n, không quá 10 chữ số.

Hãy thực hiện đếm số lần xuất hiện của các chữ số nguyên tố trong n và in ra màn hình. (Liệt kê theo thứ tự xuất hiện các chữ số)

**Input**

Chỉ có một số nguyên dương N không quá 10 chữ số.

**Output**

Ghi ra kết quả, mỗi dòng ghi một số nguyên tố và số lần xuất hiện theo thứ tự xuất hiện.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 112345 | 2 1  3 1  5 1 |

#include<stdio.h>

#include<math.h>

int b[1001];

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

long n;

scanf("%ld",&n);

long x=n;

int m=0,i;

while(n>0){

m++;

n=n/10;

}

int a[m],j;

for(i=m;i>0;i--){

a[i]=x%10;

x=x/10;

if(nt(a[i])==1){

b[a[i]]++;

}

}

int kt;

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

if(nt(a[i])==1){

kt=0;

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

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

kt=1;

break;

}

}

if(kt==0) printf("%d %d\n",a[i],b[a[i]]);

}

}

}

SẮP XẾP CHỌN - 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04050/top)

Viết chương trình C cho phép nhập vào mảng A gồm n phần tư (1<n<100). Thực hiện sắp xếp tăng dần các phần tử của mảng theo thuật toán chọn và In ra.  
Trong đó:  
INPUT  
Hàng thứ nhất là số phần tử n của mảng A  
Hàng thứ hai là các phần tử của mảng A  
OUTPUT  
Kết quả các bước sắp xếp theo thuật toán  
INPUT  
5  
64 25 12 22 11  
OUTPUT  
11 25 12 22 64  
11 12 25 22 64  
11 12 22 25 64  
11 12 22 25 64

#include<stdio.h>

int main(){

int n,i,j,k;

scanf("%d",&n);

int a[n];

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

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

}

int x;

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

int m=a[i],vt=i;

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

if(a[j]<m){

m=a[j];

vt=j;

}

}

a[vt]=a[i];

a[i]=m;

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

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

}

printf("\n");

}

}

ƯỚC SỐ CHIA HẾT CHO 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01052/top)

Cho số nguyên dương N.

Nhiệm vụ của bạn là hãy xác định xem có bao nhiêu ước số của N chia hết cho 2?

**Input:**

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

Mỗi bộ test gồm một số nguyên N (1 ≤ N ≤ 109)

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 2  9  8 | 0  3 |

#include<stdio.h>

#include<math.h>

int main(){

int t,k;

scanf("%d",&t);

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

long n,d=0,i;

scanf("%ld",&n);

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

if(n%i==0 ){

if(i%2==0) d++;

if((n/i)%2==0 && i\*i!=n) d++;

}

}

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

}

}

ƯỚC SỐ NGUYÊN TỐ LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03015/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 nt(long a){

long i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int main(){

int t,k;

scanf("%d",&t);

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

long n,i,max=0;

scanf("%ld",&n);

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

if(n%i==0){

if (nt(i)==1){

if(i>max)

max=i;

}

if(nt(n/i)==1){

if(n/i>max) max=n/i;

}

}

}

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

}

}

KIỂM TRA SỐ FIBONACCI 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03016/top)

Cho số nguyên dương n. Hãy kiểm tra xem n có phải là số trong dãy Fibonacci hay không?

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương n.
* T, n thỏa mãn ràng buộc :1 ≤ T ≤ 100; 1≤n≤1018.

**Output:**

* Đưa ra “YES” hoặc “NO” tương ứng với n là số Fibonacci hoặc không phải số Fibonacci của mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  8  15 | YES  NO |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

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

long long n;

scanf("%lld",&n);

long long i,kt=0,x,y,z;

if(n==0 || n==1) printf("YES");

else{

x=0;

y=1;

z=0;

while(z<n){

z=x+y;

x=y;

y=z;

}

if(z==n) printf("YES");

else printf("NO");

}

printf("\n");

}

}

TỔNG ƯỚC SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01054/top)

Cho N số nguyên. Nhiệm vụ của bạn là phân tích các số nguyên đã cho dưới dạng tích của các thừa số nguyên tố, sau đó tính tổng các ước số nguyên tố này.

**Input:**

* Dòng đầu tiên số nguyên N (1 ≤ N ≤ 106).
* N dòng tiếp theo, mỗi dòng gồm một số nguyên có giá trị không vượt quá 2\*106.

**Output:**

In ra một số nguyên là đáp án tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 5  7  9  10  13  100 | 47 |

#include<stdio.h>

#include<math.h>

int b[2\*1000000+9]={0};

int a[1000009];

void sangnt(){

int i,j;

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

if(b[i]==0){

b[i]=i;

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

b[j]=i;

}

}

}

}

int main(){

int t;

scanf("%d",&t);

sangnt();

long s=0;

while(t--){

long n;

scanf("%ld",&n);

while(n>1){

s=s+b[n];

n=n/b[n];

}

}

printf("%ld",s);

}

SỐ TĂNG GIẢM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03017/top)

Một số được gọi là số tăng giảm nếu số đó có các chữ số thỏa mãn hoặc tăng dần, hoặc giảm dần từ trái qua phải. Hãy đếm các số nguyên tố là số tăng giảm với **số chữ số cho trước**.

**Dữ liệu vào:**Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10)

**Kết quả:** Ghi ra số lượng các số thỏa mãn điều kiện.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  4 | 20  50 |

#include<stdio.h>

#include<math.h>

int snt(long a){

if(a<2) return 0;

else {

int kt=1;

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

if(a%i==0){

kt=0;

break;

}

}

if(kt==1) return 1;

else return 0;

}

}

int stg(long a){

int k=1,e,d=a%10;

a=a/10;

e=a%10;

int n=d,m,x=d,y;

long b=a;

if(d<e){

while(a>0){

m=a%10;;

if(n>=m){

k=0;

break;

}

n=m;

a=a/10;

}

}

else {

while(b>0){

y=b%10;

if(x<=y){

k=0;

break;

}

x=y;

b=b/10;

}

}

if(k==1) return 1;

else return 0;

}

int main(){

int x;

scanf("%d",&x);

while(x--){

int t;

scanf("%d",&t);

long n=1,m;

while(t--){

n=n\*10;

}

m=n/10;

long d=0,i;

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

if(stg(i)==1){

if(snt(i)==1)

d++;

}

}

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

}

}

SỐ ĐẸP 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C03019/top)

Một số được coi là đẹp nếu nếu nó có tính chất thuận nghịch và tổng chữ số chia hết cho 10.  Bài toán đặt ra là cho trước số chữ số. Hãy đếm xem có bao nhiêu số đẹp với số chữ số như vậy.

**Dữ liệu vào:**Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10).

**Kết quả:**Mỗi bộ test viết ra số lượng số đẹp tương ứng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  5 | 1  90 |

#include<stdio.h>

int stn(long a){

int d=0,s=0,m=a;

while(a>0){

d=d\*10+a%10;

s=s+a%10;

a=a/10;

}

if(d==m && s%10==0) return 1;

else return 0;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

int a;

scanf("%d",&a);

long n=1,m;

while(a--){

n=n\*10;

}

m=n/10;

long d=0;

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

if(stn(i)==1) d++;

}

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

}

}

SỐ XUẤT HIỆN NHIỀU LẦN NHẤT TRONG DÃY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04019/top)

Cho một dãy số nguyên dương không quá 100 phần tử, các giá trị trong dãy không quá 30000. Hãy xác định xem số nào là số xuất hiện nhiều lần nhất trong dãy. Chú ý: trong trường hợp nhiều số khác nhau cùng xuất hiện số lần bằng nhau và là lớn nhất thì in ra tất cả các số đó theo thứ tự xuất hiện trong dãy ban đầu.

**Input:**Dòng đầu là số bộ test, không quá 20. Mỗi bộ test gồm hai dòng. Dòng đầu ghi số phần tử của dãy, dòng tiếp theo ghi các phần tử của dãy.

**Output:**Với mỗi bộ test, đưa ra số xuất hiện nhiều lần nhất trong dãy đã cho.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  10  1 2 3 1 2 3 1 2 3 1  10  1 2 3 4 5 6 7 8 9 0 | 1  1 2 3 4 5 6 7 8 9 0 |

#include<stdio.h>

int b[40000];

int main(){

int t,k;

scanf("%d",&t);

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

int n;

scanf("%d",&n);

int a[1000],i,j,max=0;

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

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

b[a[i]]++;

if(b[a[i]]>max){

max=b[a[i]];

}

}

int kt;

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

if(b[a[i]]==max){

kt=1;

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

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

kt=0;

break;

}

}

}

if(b[a[i]]==max && kt==1) printf("%d ",a[i]);

}

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

b[a[i]]=0;

}

printf("\n");

}

}

ĐOẠN TĂNG DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04031/top)

Một đoạn tăng trong một dãy số nguyên là một đoạn liên tiếp trong dãy sao cho phần từ phía sau lớn hơn phần từ phía trước. Cho dãy số với n phần tử (n không quá 100, các phần tử đều không quá 1000). Viết chương trình tìm các đoạn tăng liên tiếp trong dãy mà số phần tử là nhiều nhất.

**Input:** Dòng đầu ghi số bộ test. Mỗi test gồm 2 dòng, dòng đầu ghi số N là số phần tử của dãy. Dòng sau ghi N số của dãy. N không quá 100, các số trong dãy đều nguyên dương và không quá 1000.

**Output:** Với mỗi bộ test, ghi ra thứ tự bộ test. Sau đó là 1 dòng ghi độ dài của đoạn tăng dài nhất. Tiếp theo là một số dòng ghi lần lượt các đoạn tăng dài nhất, từ trái qua phải trong dãy ban đầu.

**Ví dụ:**

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

#include<stdio.h>

int b[100001]={0};

int main(){

int t,k=1;

scanf("%d",&t);

while(t--){

int n,d=1,max=1;

scanf("%d",&n);

int a[n+1],i;

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

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

b[i]=1;

}

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

if(a[i-1]<a[i]) d++;

else {

d=1;

}

b[i]=d;

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

}

printf("Test %d:\n%d\n",k,max);

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

if(b[i]==max){

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

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

if(j==i) printf("\n");

}

}

}

k++;

}

}

TRỘN HAI DÃY VÀ SẮP XẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04030/top)

Cho hai dãy số nguyên dương A và B không quá 100 phần tử, các giá trị trong dãy không quá 30000 và số phần tử của hai dãy bằng nhau.  Hãy trộn hai dãy với nhau sao cho dãy A được đưa vào các vị trí có chỉ số chẵn, dãy B được đưa vào các vị trí có chỉ số lẻ. Đồng thời, dãy A được sắp xếp tăng dần, còn dãy B được sắp xếp giảm dần.     (Chú ý: chỉ số tính từ 0)

**Input:**Dòng 1 ghi số bộ test. Với mỗi bộ test:  dòng đầu tiên ghi số n. Dòng tiếp theo ghi n số nguyên dương của dãy A.  Dòng tiếp theo ghi n số nguyên dương của dãy B

**Output:**Với mỗi bộ test, đưa ra thứ tự bộ test và dãy kết quả.

**Ví dụ:**

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

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

int k=0;

while(t--){

k++;

int n,i,j;

scanf("%d",&n);

long a[n],b[n],min,max;

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

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

}

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

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

}

long d,e;

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

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

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

d=a[j];

a[j]=a[i];

a[i]=d;

}

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

e=b[j];

b[j]=b[i];

b[i]=e;

}

}

}

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

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

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

}

printf("\n");

}

}

SẮP XẾP ĐỔI CHỖ TRỰC TIẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04026/top)

Hãy thực hiện thuật toán sắp xếp đổi chỗ trực tiếp 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 7 3 2 | Buoc 1: 2 7 5 3  Buoc 2: 2 3 7 5  Buoc 3: 2 3 5 7 |

#include<stdio.h>

int main(){

int n,i,j,k;

scanf("%d",&n);

int a[n];

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

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

}

int x;

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

printf("Buoc %d: ",i+1);

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

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

int d=a[j];

a[j]=a[i];

a[i]=d;

}

}

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

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

}

printf("\n");

}

}

SẮP XẾP CHỌN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04027/top)

Hãy thực hiện thuật toán sắp xếp chọn 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 7 3 2 | Buoc 1: 2 7 3 5  Buoc 2: 2 3 7 5  Buoc 3: 2 3 5 7 |

#include<stdio.h>

int main(){

int n,i,j,k;

scanf("%d",&n);

int a[n];

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

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

}

int x;

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

int m=a[i],vt=i;

printf("Buoc %d: ",i+1);

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

if(a[j]<m){

m=a[j];

vt=j;

}

}

a[vt]=a[i];

a[i]=m;

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

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

}

printf("\n");

}

}

SẮP XẾP CHÈN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04028/top)

Hãy thực hiện thuật toán sắp xếp chèn 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 7 3 2 | Buoc 0: 5  Buoc 1: 5 7  Buoc 2: 3 5 7  Buoc 3: 2 3 5 7 |

#include<stdio.h>

int main(){

int n,i,j,k;

scanf("%d",&n);

int a[n];

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

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

}

int x;

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

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

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

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

int d=a[j];

a[j]=a[i];

a[i]=d;

}

}

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

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

}

printf("\n");

}

}

ĐỔI TIỀN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04036/top)

Tại ngân hàng có các mệnh giá bằng 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, số lượng tờ tiền mỗi mệnh giá là không hạn chế. Một người cần đổi số tiền có giá trị bằng N. Hãy xác định xem số tờ tiền ít nhất sau khi đổi là bao nhiêu?

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 50). Mỗi test gồm 1 số nguyên N ( 1 ≤ N ≤ 100000).

**Output:**Với mỗi test, in ra đáp án trên một dòng.

**Test ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  70  121 | 2  3 |

#include<stdio.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

int n,d=0;

scanf("%d",&n);

d=n/1000;

n=n-(n/1000)\*1000;

d=d+n/500;

n=n-(n/500)\*500;

d=d+n/200;

n=n-(n/200)\*200;

d=d+n/100;

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

d=d+n/50;

n=n-(n/50)\*50;

d=d+n/20;

n=n-(n/20)\*20;

d=d+n/10;

n=n-(n/10)\*10;

d=d+n/5;

n=n-(n/5)\*5;

d=d+n/2;

n=n-(n/2)\*2;

d=d+n;

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

}

}

 TÍCH THỪA SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C01032/top)

Cho một số nguyên dương không quá 9 chữ số. Người ta phân tích số đó thành tích các thừa số nguyên tố sau đó tính lại một giá trị mới bằng cách nhân các thừa số nguyên tố khác nhau của số đó.

Ví dụ: Số 72 được phân tích thành 23 \* 32. Giá trị tính được sẽ lã 2 \* 3 = 6

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test là một số nguyên dương không quá 109

**Kết quả**

* Với mỗi bộ test, ghi ra kết quả tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  72  1000  997 | 6  10  997 |

#include<stdio.h>

int main(){

int t,k;

scanf("%d",&t);

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

long i, n,s=1;

scanf("%ld",&n);

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

if(n%i==0) s=s\*i;

while(n%i==0){

n=n/i;

}

}

printf("%ld",s);

printf("\n");

}

}

LOẠI BỎ HÀNG VÀ CỘT CÓ TỔNG LỚN NHẤT RA KHỎI MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05008/top)

Cho một ma trận cấp M\*N chỉ bao gồm các số nguyên dương không quá 1000. Hãy viết chương trình trước hết loại bỏ hàng có tổng lớn nhất sau đó tính toán lại để loại tiếp cột có tổng lớn nhất ra khỏi ma trận.

**Input:**Dòng đầu tiên ghi số bộ test.  Mỗi bộ test viết trên một dòng hai số N và M lần lượt là số hàng và số cột của ma trận (2<N,M<10). Tiếp thep là N dòng ghi các số của ma trận.

**Output:**Mỗi bộ test viết ra thứ tự bộ test, sau đó là ma trận sau khi tính toán.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 3 3 1 2 4 3 4 0 6 3 5 | Test 1:  1 4 3 0 |

#include<stdio.h>

int main(){

int t,x=1;

scanf("%d",&t);

while(t--){

int m,n,i,j;

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

int a[m][n];

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

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

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

}

}

int max=a[0][0],k=a[0][0];

int s,ss,vt,d;

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

s=0;

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

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

}

if(s>max){

max=s;

vt=i;

}

}

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

ss=0;

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

if(i!=vt){

ss=ss+a[i][j];

}

}

if(ss>k){

k=ss;

d=j;

}

}

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

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

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

if(i!=vt && j!=d){

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

}

}

printf("\n");

}

x++;

}

}

TÍCH MA TRẬN VỚI CHUYỂN VỊ CỦA NÓ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05011/top)

Cho ma trận A chỉ gồm các số nguyên dương cấp N\*M. Hãy viết chương trình tính tích của A với ma trận chuyển vị của A.

**Input:**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.

**Output:**Với mỗi bộ test ghi ra thứ tự bộ test, sau đó đến ma trận tích tương ứng, mỗi số cách nhau đúng một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2  2  1  2  3  4 | Test 1:  5 11  11 25 |

#include<stdio.h>

int main(){

int t,x=1;

scanf("%d",&t);

while(t--){

int n,m,i,j,k;

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

int a[n][m];

int b[m][n];

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

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

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

}

}

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

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

b[i][j]=a[j][i];

}

}

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

int c[n][n];

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

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

{

c[i][k] = 0;

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

c[i][k] = c[i][k] + a[i][j]\*b[j][k];

}

printf("%d ",c[i][k]);

}

printf("\n");

}

x++;

}

}

TÍCH MA TRẬN CHÉO

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05012/top)

Cho một số nguyên dương N không quá 20. Ma trận vuông A cấp N\*N được tạo theo mẫu trong bảng dưới. Viết chương trình tính tích của A với chuyển vị của A.

|  |  |
| --- | --- |
| Với N = 4 1 0 0 0 1 2 0 0 1 2 3 0 1 2 3 4 | Với N = 5 1 0 0 0 0 1 2 0 0 0 1 2 3 0 0 1 2 3 4 0 1 2 3 4 5 |

#include<stdio.h>

int main(){

int t,x=1;

scanf("%d",&t);

while(t--){

int n,i,j,k;

scanf("%d",&n);

int a[n][n];

int b[n][n];

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

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

a[i][j]=j+1;

}

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

a[i][j]=0;

}

}

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

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

b[i][j]=a[j][i];

}

}

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

int c[n][n];

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

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

{

c[i][k] = 0;

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

c[i][k] = c[i][k] + a[i][j]\*b[j][k];

}

printf("%d ",c[i][k]);

}

printf("\n");

}

x++;

}

}

MA TRẬN XOÁY ỐC NGƯỢC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05018/top)

Ma trận xoáy ốc ngược cấp N là ma trận vuông có N\*N phần tử. Các số được điền vào ma trận trận theo chiều kim đồng hồ theo thứ tự giảm dần về 1.

**Dữ liệu vào**

* Dòng 1 ghi số bộ test
* Mỗi bộ test ghi số N (1<N<20).

**Kết quả**

            Ghi ra thứ tự bộ test và ma trận xoáy ốc ngược tương ứng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  3 | Test 1:  9 8 7  2 1 6  3 4 5 |

#include<stdio.h>

int main(){

int t,y=1;

scanf("%d",&t);

while(t--){

int n;

scanf("%d",&n);

int m=n;

int x=n\*n,k=0,i,j;

int a[n][n];

while(x>0){

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

a[k][j]=x--;

}

for(i=k+1;i<m;i++){

a[i][m-1]=x--;

}

for(j=m-2;j>=k;j--){

a[m-1][j]=x--;

}

for(i=m-2;i>k;i--){

a[i][k]=x--;

}

m--;

k++;

}

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

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

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

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

}

printf("\n");

}

y++;

}

}

MA TRẬN XOÁY ỐC NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C05019/top)

Ma trận xoáy ốc nguyên tố cấp N là ma trận vuông có N\*N phần tử. Các số được điền vào ma trận theo chiều kim đồng hồ đều là các số nguyên tố từ nhỏ đến lớn.

**Dữ liệu vào**

* Dòng 1 ghi số bộ test
* Mỗi bộ test ghi số N (1<N<20).

**Kết quả**

            Ghi ra thứ tự bộ test và ma trận xoáy ốc nguyên tố tương ứng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  3 | Test 1:   2  3  5  19 23  7  17 13 11 |

#include<stdio.h>

#include<math.h>

int nt(int a){

int i,kt=0;

if(a<=1) return 0;

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

if(a%i==0){

kt=1;

break;

}

}

if(kt==1) return 0;

else return 1;

}

int ntt(int n){

int i=1;

int d=0,x;

while(i++){

if(nt(i)==1){

d++;

x=i;

}

if(d==n){

break;

}

}

return x;

}

int main(){

int t;

scanf("%d",&t);

int o=1;

while (t--){

int n;

scanf("%d",&n);

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

int x=1,k=0,i,j;

int a[n][n];

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

while(x<=p){

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

a[k][j]=ntt(x);

x++;

}

for(i=k+1;i<m;i++){

a[i][m-1]=ntt(x);

x++;

}

for(j=m-2;j>=k;j--){

a[m-1][j]=ntt(x);

x++;

}

for(i=m-2;i>k;i--){

a[i][k]=ntt(x);

x++;

}

m--;

k++;

}

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

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

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

}

printf("\n");

}

o++;

printf("\n");

}

}

CHUẨN HÓA XÂU HỌ TÊN 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06015/top)

Các cán bộ, giảng viên PTIT khi tham gia hội nghị quốc tế sẽ được viết lại xâu họ tên theo dạng chuẩn trong đó họ được viết sau cùng, phân tách với phần tên đệm và tên bởi dấu phẩy. Các chữ cái của họ đều viết hoa.

Cho trước các xâu họ tên (có thể không chuẩn). Hãy đưa về dạng chuẩn 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 xâu chuẩn.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4      nGUYEn    quaNG   vInH     tRan   thi THU    huOnG     nGO   quoC  VINH   lE            tuAn    aNH | Quang Vinh, NGUYEN  Thi Thu Huong, TRAN  Quoc Vinh, NGO  Tuan Anh, LE |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

char a[50];

scanf("\n");

gets(a);

int x=0 ,j=0 , l=strlen(a), vt ,i ;

while(a[x]==' ') {

x++;

j++;

}

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

l--;

}

while(x<l){

if(a[x]==' ') {

vt=x;

break;

}

x++;

}

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

if(a[i-1]==' ' && a[i]!=' '){

if(a[i]>=97 && a[i]<=122) {

a[i]=a[i]-32;

}

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

}

else if(a[i-1]!=' ' && a[i]!=' ' && a[i+1]!=' '){

if(a[i]>=65 && a[i]<=90) {

a[i]=a[i]+32;

}

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

}

else if( a[i-1]!=' ' && a[i]!=' ' && a[i+1]==' '){

if(a[i]>=65 && a[i]<=90){

a[i]=a[i]+32;

}

if(i!=l-1) {

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

printf(" ");

}

else printf("%c",a[i]);

}

}

printf(", ");

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

if(a[i]>=97 && a[i]<=122){

a[i]=a[i]-32;

}

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

}

printf("\n");

}

}

TẬP TỪ RIÊNG CỦA HAI XÂU 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06018/top)

Cho hai xâu ký tự S1 và S2. Hãy viết chương trình tìm các từ chỉ xuất hiện trong S1 mà không xuất hiện trong S2. Chú ý: mỗi từ chỉ liệt kê 1 lần.

**Input:**Dòng 1 ghi số bộ test. Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một xâu ký tự độ dài không quá 200, chỉ bao gồm các ký tự viết thường và các khoảng trống.

**Output:**          Với mỗi bộ test ghi ra các từ có trong S1 mà không có trong S2. Các từ được ghi theo thứ tự từ điển.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  abc ab ab ab abcd  ab abc  aaa xyz ab zzz abc dd dd abc  xyz dd ttt sas cdc | abcd  aaa ab abc zzz |

#include<stdio.h>

#include<string.h>

int main(){

int T;

scanf("%d",&T);

while(T--){

scanf("\n");

char a[202],b[202],s1[202][50],s2[202][50];

gets(a);

scanf("\n");

gets(b);

int la=strlen(a), lb=strlen(b), i, j,n=0,m=0;

char \*p1=strtok(a," ");

while(p1!=NULL){

strcpy(s1[n],p1);

n++;

p1=strtok(NULL," ");

}

char \*p2=strtok(b," ");

while(p2!=NULL){

strcpy(s2[m],p2);

m++;

p2=strtok(NULL," ");

}

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

if(a[i]!=' ') s1[n][j++]=a[i];

if(a[i]==' ' || i==la-1) {

n++;

j=0;

}

}

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

if(b[i]!=' ') s2[m][k++]=b[i];

if(b[i]==' ' || i==lb-1) {

m++;

k=0;

}

}\*/

char t[50];

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

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

if(strcmp(s1[i],s1[j])>0){

strcpy(t,s1[j]);

strcpy(s1[j],s1[i]);

strcpy(s1[i],t);

}

}

}

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

int kt=0;

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

if(strcmp(s1[i],s1[j])==0){

kt=1;

break;

}

}

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

if(strcmp(s1[i],s2[j])==0){

kt=1;

break;

}

}

if(kt==0) printf("%s ",s1[i]);

}

printf("\n");

}

}

XÓA TỪ TRONG XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06022/top)

Cho trước một xâu ký tự S1 chỉ bao gồm các chữ cái và khoảng trống cùng một từ S2. Hãy tìm xem S2 có xuất hiện trong S1 hay không. Nếu có loại bỏ tất cả những lần xuất hiện của S2 trong S1. Chú ý: tìm S2 trong S1 theo kiểu không phân biệt chữ hoa chữ thường

**Input:**Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên hai dòng: Dòng đầu ghi xâu ký tự S1, độ dài không quá 200. Dòng thứ 2 ghi từ S2 (không quá 20 ký tự)

**Output:** Với mỗi bộ test ghi ra thứ tự bộ test và xâu kết quả sau khi đã xóa.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Abc ddd abdc aaa bbb abc ddD XY aBc ACHDNC XXXX YYYY ABC ABC XXXX XxXx | Test 1: ddd abdc aaa bbb ddD XY Test 2: ACHDNC YYYY ABC ABC |

**#include<stdio.h>**

**#include<string.h>**

**int main(){**

**int t,o=1;**

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

**while(t--){**

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

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

**gets(a);**

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

**gets(b);**

**int m=strlen(a), n=strlen(b),i,j;**

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

**int k=1;**

**while(k!=0){**

**if((a[i+k-1]!=b[k-1]) && (a[i+k-1]!=b[k-1]+32) && (a[i+k-1]!=b[k-1]-32)) break;**

**else k++;**

**if(k==n+1){**

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

**a[j]='\0';**

**}**

**k=0;**

**}**

**}**

**}**

**printf("Test %d: ",o);**

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

**if(a[i]!='\0') printf("%c",a[i]);**

**}**

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

**o++;**

**}**

**}**

 XÂU ĐỐI XỨNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06035/top)

Cho trước một xâu S.  Bạn hãy thay đổi đúng 1 kí tự, sao cho xâu mới thu được là một xâu đối xứng.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤  20). Mỗi test ghi trên một dòng 1 xâu S có độ dài không quá 20 kí tự.

**Output:**Với mỗi test, in ra “YES” nếu có thể biến đổi xâu S thành xâu đối xứng, in ra “NO” trong trường hợp ngược lại.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  abccaa  abbcca  abcda | YES  NO  YES |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

char a[21];

scanf("\n");

gets(a);

int i=0,l=strlen(a);

int d=0,n=l/2;

if(l%2==0){

while(i<n){

if(a[i]!=a[l-1]) {

d++;

}

l--;

i++;

}

if(d==1) printf("YES\n");

else printf("NO\n");

}

else {

while(i<n){

if(a[i]!=a[l-1]) {

d++;

}

l--;

i++;

}

if(d<=1) printf("YES\n");

else printf("NO\n");

}

}

}

CHUẨN HÓA XÂU HỌ TÊN 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06014/top)

Một xâu họ tên được coi là viết chuẩn nếu chữ cái đầu tiên mỗi từ được viết hoa, các chữ  
cái khác viết thường. Các từ cách nhau đúng một dấu cách và không có khoảng trống  
thừa ở đầu và cuối xâu. Hãy viết chương trình đưa các xâu họ tên về dạng chuẩn.  
**Input:**  
Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng xâu ký tự họ tên, không quá  
80 ký tự.  
**Output:**  
Với mỗi bộ test ghi ra xâu ký tự họ tên đã chuẩn hóa.  
**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3    nGuYEN    vAN    naM tRan           TRUNG hiEU       vO le         hOA bINh | Nguyen Van Nam Tran Trung Hieu Vo Le Hoa Binh |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

char a[50];

scanf("\n");

gets(a);

int x=0 ,j=0 , l=strlen(a), vt ,i ;

while(a[x]==' ') {

x++;

j++;

}

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

l--;

}

if(a[x]>=97 && a[x]<=122){

a[x]=a[x]-32;

printf("%c",a[x]);

}

for(i=x+1;i<l;i++){

if(a[i-1]==' ' && a[i]!=' '){

if(a[i]>=97 && a[i]<=122) {

a[i]=a[i]-32;

}

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

}

else if(a[i-1]!=' ' && a[i]!=' ' && a[i+1]!=' '){

if(a[i]>=65 && a[i]<=90) {

a[i]=a[i]+32;

}

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

}

else if( a[i-1]!=' ' && a[i]!=' ' && a[i+1]==' '){

if(a[i]>=65 && a[i]<=90){

a[i]=a[i]+32;

}

if(i!=l-1) {

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

printf(" ");

}

else printf("%c",a[i]);

}

}

printf("\n");

}

}

TÌM TỪ THUẬN NGHỊCH DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06026/top)

Cho một văn bản. Hãy tìm ra từ thỏa mãn tính chất **thuận nghịch có độ dài lớn nhất**trong và cho biết từ đó **xuất hiện bao nhiêu lần**. Nếu có nhiều từ cùng có độ dài lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện.

**Input:**Văn bản không quá 1000 từ.

**Output:**

Ghi ra trên một dòng từ thuận nghịch có độ dài lớn nhất và số lần xuất hiện của nó. Nếu có nhiều từ cùng có độ dài lớn nhất thì các từ được liệt kê theo thứ tự xuất hiện ban đầu.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| AAA BAABA HDHDH ACBSD SRGTDH DDDDS DUAHD AAA AD DA HDHDH AAA AAA AAA AAA DDDAS HDHDH HDH AAA AAA AAA AAA AAA AAA AAA AAA DHKFKH DHDHDD HDHDHD DDDHHH HHHDDD TDTD | HDHDH 3 |

#include<stdio.h>

#include<string.h>

int ktra(char a[]){

int l=strlen(a);

for(int i=0;i<l/2;i++){

if(a[i]!=a[l-1-i]) return 0;

}

return 1;

}

struct chuoi{

char xau[101];

};

int main(){

struct chuoi s[8000];

scanf(" ");

int n=0;

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

n++;

}

int max=0;

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

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

if(l>max && ktra(s[i].xau) )max=l;

}

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

int kt=0,dem=1;

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

if(l==max && ktra(s[i].xau)==1){

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\n",s[i].xau,dem);

}

}

}

}

SỐ ĐẦY ĐỦ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06013/top)

Cho một số nguyên dương lớn có nhiều hơn 20 chữ số nhưng không quá 1000 chữ số. Hãy kiểm tra xem số đó có đầy đủ tất cả các chữ số từ 0 đến 9 hay không.

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test là một dãy ký tự có độ dài không quá 1000, không có khoảng trống

**Kết quả**

* Nếu dữ liệu vào không phải là một số nguyên hợp lệ (có ký tự không phải số hoặc bắt đầu bằng chữ số 0) thì in ra INVALID
* Nếu dữ liệu vào thỏa mãn đầy đủ thì in ra YES, nếu không in ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  01234aa32432432432534545b987978  123444444444444566666666667890  324562783924723543243243242354354354333234324 | INVALID  YES  NO |

#include<stdio.h>

#include<string.h>

int main(){

char a[1001];

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

gets(a);

int i,l=strlen(a),d0=0,d1=0,d2=0,d3=0,d4=0,d5=0,d6=0,d7=0,d8=0,d9=0,kt=1,d=0;

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

d++;

}

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

l--;

}

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

if((a[i]<'0' || a[i] > '9') || a[d]=='0') {

printf("INVALID");

kt=0;

break;

}

else

{

if(a[i]=='0') d0++;

else if(a[i]=='1') d1++;

else if(a[i]=='2') d2++;

else if(a[i]=='3') d3++;

else if(a[i]=='4') d4++;

else if(a[i]=='5') d5++;

else if(a[i]=='6') d6++;

else if(a[i]=='7') d7++;

else if(a[i]=='8') d8++;

else d9++;

}

}

if(kt==1){

if( d0>0 && d1>0 && d2>0 && d3>0 && d4>0 && d5>0 && d6>0 && d7>0 && d8>0 && d9>0) printf("YES");

else printf("NO");

}

printf("\n");

}

}

SỐ ƯU THẾ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06021/top)

Cho một số nguyên dương lớn có nhiều hơn 20 chữ số nhưng không quá 1000 chữ số. Một số nguyên dương được coi là “số ưu thế chẵn” nếu số chữ số của nó là chẵn và số chữ số chẵn nhiều hơn số chữ số lẻ.

Một số nguyên dương được coi là “số ưu thế lẻ” nếu số chữ số của nó là lẻ và số chữ số lẻ nhiều hơn số chữ số chẵn.

Hãy kiểm tra xem số đó có phải là số ưu thế (chẵn hoặc lẻ) hay không. Chú ý: trường hợp số lượng chữ số chẵn và số lượng chữ số lẻ bằng nhau thì không được coi là số ưu thế.

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test là một dãy ký tự có độ dài không quá 1000, không có khoảng trống

**Kết quả**

* Nếu dữ liệu vào không phải là một số nguyên hợp lệ (có ký tự không phải số hoặc bắt đầu bằng chữ số 0) thì in ra INVALID
* Nếu dữ liệu vào thỏa mãn là số ưu thế thì in ra YES, nếu không in ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  01234aa32432432432534545b987978  1234567890123456789000  999999999999999999999999999999 | INVALID  YES  NO |

#include<stdio.h>

#include<string.h>

int main(){

char a[1001];

int t;

scanf("%d",&t);

while(t--){

scanf("\n");

gets(a);

int i,l=strlen(a),d0=0,d1=0,kt=1,d=0;

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

d++;

}

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

l--;

}

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

if((a[i]<'0' || a[i] > '9') || a[d]=='0') {

printf("INVALID");

kt=0;

break;

}

else

{

if((l-d)%2!=0) {

printf("NO");

kt=0;

break;

}

else {

if(a[i]=='0' || a[i]=='2' || a[i]=='4'|| a[i]=='6' || a[i]=='8') d0++;

else d1++;

}

}

}

if(kt==1){

if( d0>d1) printf("YES");

else printf("NO");

}

printf("\n");

}

}

BIỂN SỐ ĐẸP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C06009/top)

Biển số xe máy được quy định gồm các thành phần:

* Hai chữ số đầu là mã quản lý theo tỉnh – thành phố (ví dụ mã của Hà Nội là 29 đến 31)
* Tiếp theo là dấu gạch ngang, sau đó là một chữ cái in hoa (từ A đến Z) và một chữ số. Cặp chữ cái và chữ số này được cấp theo khu vực quận – huyện.
* Tiếp theo là một dãy 5 số gồm 2 cụm: 3 chữ số đầu và hai chữ số sau, phân tách bởi dấu chấm.

Thông thường, người ta chỉ quan tâm đến 5 chữ số cuối. Giả sử các trường hợp sau được coi là đẹp:

* Cả 5 chữ số được sắp theo thứ tự tăng chặt từ trái qua phải.
* Cả 5 chữ số đều bằng nhau
* Ba chữ số đầu bằng nhau và hai chữ số cuối bằng nhau
* Cả 5 chữ số đều là 6 và/hoặc 8 (số lộc phát).

Theo quy tắc trên, các biển số sau được coi là đẹp:

* 29-T1 123.79
* 29-T1 555.55
* 29-T1 222.33
* 29-T1 686.88

Và các biển số sau không đẹp:

* 29-T1 123.33
* 29-T1 555.54
* 29-T1 606.88

Viết chương trình kiểm tra xem các biển số xe có đẹp hay không.

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 50
* Mỗi bộ test là một biển số. Dữ liệu vào đảm bảo biển số được viết đúng quy định.

**Kết quả**

* In ra kết quả kiểm tra với từng bộ test

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7  29-T1 123.45  29-T1 555.55  29-T1 222.33  29-T1 686.88  29-T1 123.33  29-T1 555.54  29-T1 606.88 | YES  YES  YES  YES  NO  NO  NO |

#include<stdio.h>

#include<string.h>

int main(){

int t;

scanf("%d",&t);

while(t--){

char a[100];

scanf("\n");

gets(a);

int l=strlen(a);

int i;

int kt=0;

if(a[6]<a[7] && a[7]<a[8] && a[8]<a[10] && a[10]<a[11]){

kt=1;

}

else if(a[6]==a[7] && a[7]==a[8] && a[8]==a[10] && a[10]==a[11]){

kt=1;

}

else if(a[6]==a[7] && a[7]==a[8] && a[10]==a[11]){

kt=1;

}

else if( (a[6]==54 || a[6]==56) && (a[7]==54 || a[7]==56) && (a[8]==54 || a[8]==56) && (a[10]==54 || a[10]==56) && (a[11]==54 || a[11]==56)){

kt=1;

}

else {

kt=0;

}

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

else printf("YES\n");

}

}

 DÃY ƯU THẾ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C04045/top)

Cho dãy A[] chỉ bao gồm các số nguyên dương không quá 105 nhưng không biết trước số phần tử của dãy. Người ta gọi dãy A[] là dãy ưu thế nếu thỏa mãn 1 trong 2 điều kiện sau đây:

* Dãy gọi là ưu thế chẵn nếu số phần tử của dãy là chẵn và số lượng số chẵn trong dãy nhiều hơn số lượng số lẻ.
* Dãy gọi là ưu thế lẻ nếu số phần tử của dãy là lẻ và số lượng số lẻ trong dãy nhiều hơn số lượng số chẵn.

Hãy kiểm tra xem dãy A[] có phải là dãy ưu thế hay không.

**Dữ liệu vào**

* Dòng đầu ghi số bộ test, không quá 10
* Mỗi bộ test là một dãy các số nguyên dương (không quá 104) và có không quá 200 số, các số cách nhau 1 khoảng trống, không biết trước số lượng phần tử.

**Kết quả**

* Nếu dãy A[] thỏa mãn là dãy ưu thế thì in ra YES, nếu không in ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  11 22 33 44 55 66 77  23 34 45 56 67 78 89 90 121 131 141 151 161 171 | YES  NO |

#include<stdio.h>

#include<string.h>

int main(){

int t;

char a[10001];

scanf("%d",&t);

while(t--){

scanf("\n");

gets(a);

int l=strlen(a),i;

int d=0,e=0,x=1;

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

if(a[i+1]==' '){

x++;

if(a[i]%2==0) d++;

else e++;

}

}

if(a[l-1]%2==0) d++;

else e++;

if(x%2==0 && d>e) printf("YES\n");

else if(x%2!=0 && d<e) printf("YES\n");

else printf("NO\n");

}

}

PHÂN SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07004/top)

Cho hai phân số p và q với tử số và mẫu số không quá 10000. Hãy viết chương trình thực hiện:

1. Rút gọn sau đó Quy đồng hai phân số
2. Tính tổng hai phân số và rút gọn
3. Tính thương hai phân số và rút gọn

**Input:** Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng 4 số nguyên dương lần lượt là tử số của p, mẫu số của p, tử số của q, mẫu số của q. Mỗi số cách nhau 1 khoảng trống

**Output:** Dòng đầu ghi dòng chữ Case và thứ tự bộ test (theo mẫu trong ví dụ). Tiếp theo là 3 dòng:

* Dòng 1 ghi 2 phân số sau khi quy đồng
* Dòng 2 ghi phân số tổng
* Dòng 3 ghi phân số thương

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2 3 4 5  1 4 7 8 | Case #1:  10/15 12/15  22/15  5/6  Case #2:  2/8 7/8  9/8  2/7 |

#include<stdio.h>

int uoc(int a,int b){

while(a%b!=0){

int d=a%b;

a=b;

b=d;

}

return b;

}

int boi(int a,int b){

return (a\*b)/(uoc(a,b));

}

int main(){

int t,a,b,c,d;

scanf("%d",&t);

int l=1;

while(t--){

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

int x=uoc(a,b),y=uoc(c,d);

a=a/x;

b=b/x;

c=c/y;

d=d/y;

int e=boi(b,d);

int x1=a\*(e/b),x2=c\*(e/d);

printf("Case #%d:\n",l);

printf("%d/%d %d/%d\n",x1,e,x2,e);

int st=(x1+x2)/uoc(x1+x2,e),sm=e/uoc((x1+x2),e);

printf("%d/%d\n",st,sm);

int tt=x1/(uoc(x1,x2)),tm=(x2/uoc(x1,x2));

printf("%d/%d\n",tt,tm);

l++;

}

}

SẮP XẾP DANH SÁCH MẶT HÀNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07012/top)

Hãy sắp xếp danh sách các mặt hàng theo lợi nhuận giảm dần. Mỗi mặt hàng gồm các thông tin: Mã mặt hàng (là một số nguyên, tự động tăng, tính từ 1); Tên mặt hàng, nhóm hàng: là các xâu ký tự; Giá mua, giá bán: là các số thực (không quá 9 chữ số)

**Input:**

Dòng đầu chứa số mặt hàng. Mỗi mặt hàng viết trên 4 dòng: Dòng 1: Tên mặt hàng. Dòng 2: Nhóm hàng. Dòng 3: Giá mua. Dòng 4: Giá bán

**Output:** Ghi ra danh sách mặt hàng đã sắp xếp theo lợi nhuận giảm dần (lợi nhuận tính bằng giá bán trừ đi giá mua). Mỗi mặt hàng viết trên một dòng gồm: mã, tên, nhóm hàng và lợi nhuận. Các thông tin cách nhau đúng 1 khoảng trống. Lợi nhuận viết với 2 chữ  số sau dấu phẩy.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  May tinh SONY VAIO  Dien tu  16400  17699  Tu lanh Side by Side  Dien lanh  18300  25999  Banh Chocopie  Tieu dung  27.5 37 | 2 Tu lanh Side by Side Dien lanh 7699.00  1 May tinh SONY VAIO Dien tu 1299.00  3 Banh Chocopie Tieu dung 9.50 |

#include<stdio.h>

#include<string.h>

struct bh{

int mmh;

char mh[1001];

char nh[1001];

float gm;

float gb;

float ln;

};

int main(){

int t,i,j;

scanf("%d",&t);

struct bh b[t+1],x;

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

scanf("\n");

gets(b[i].mh);

scanf("\n");

gets(b[i].nh);

scanf("\n");

scanf("%f",&b[i].gm);

scanf("%f",&b[i].gb);

b[i].ln=b[i].gb-b[i].gm;

b[i].mmh=i;

}

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

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

if(b[j].ln>b[i].ln){

x=b[j];

b[j]=b[i];

b[i]=x;

}

}

}

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

printf("%d %s %s %.2f\n",b[i].mmh,b[i].mh,b[i].nh,b[i].ln);

}

}

TÌM THỦ KHOA CỦA KỲ THI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07015/top)

Cho danh sách thí sinh gồm các thông tin: Mã thí sinh: là một số nguyên, tự động tăng, tính từ 1; Tên thí sinh, ngày sinh, điểm môn 1, điểm môn 2, điểm môn 3. Hãy tìm thủ khoa trong danh sách đó. Nếu có nhiều thí sinh có điểm bằng nhau và đều cao nhất thì in ra tất cả thí sinh đó theo mã tăng dần.

**Input:**

Dòng đầu chứa số thí sinh. Mỗi thí sinh viết trên 3 dòng: Dòng 1: Tên thí sinh, Dòng 2: Ngày sinh, Dòng 3,4,5: 3 điểm thi tương ứng. Các điểm thi đều đảm bảo hợp lệ (từ 0 đến 10).

**Output:** In ra các thủ khoa của kỳ thi, mỗi thí sinh 1 dòng, gồm mã, tên, ngày sinh và tổng điểm. Chú ý: nếu có nhiều thí sinh bằng điểm nhau và cao nhất thì in ra tất cả thí sinh đó theo thứ tự mã thí sinh tăng dần.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 Nguyen Van A 12/12/1994 3.5 7.0 5.5 Nguyen Van B 1/9/1994 7.5 9.5 9.5 Nguyen Van C 6/7/1994 8.5 9.5 8.5 | 2 Nguyen Van B 1/9/1994 26.5 3 Nguyen Van C 6/7/1994 26.5 |

#include<stdio.h>

struct sv{

int stt;

char ten[1001];

int ngay;

int thang;

int nam;

float d1;

float d2;

float d3;

float tong;

};

int main(){

int t,i,j;

scanf("%d",&t);

struct sv a[t+1];

float max=0;

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

scanf("\n");

gets(a[i].ten);

scanf("\n");

scanf("%d/%d/%d",&a[i].ngay,&a[i].thang,&a[i].nam);

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

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

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

a[i].stt=i;

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

if(a[i].tong>max) max=a[i].tong;

}

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

if(a[i].tong==max){

printf("%d %s %d/%d/%d %g\n",a[i].stt,a[i].ten,a[i].ngay,a[i].thang,a[i].nam,a[i].tong);

}

}

}

SẮP XẾP THÍ SINH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/C07016/top)

Hãy sắp xếp danh sách thí sinh theo tổng điểm giảm dần.

Mỗi thí sinh gồm các thông tin:

* Mã thí sinh: là một số nguyên, tự động tăng. Tính từ 1.
* Tên thí sinh, ngày sinh
* Điểm môn 1, điểm môn 2, điểm môn 3

**Input:**

Dòng đầu chứa số thí sinh. Mỗi thí sinh viết trên 3 dòng:

* Dòng 1: Tên thí sinh
* Dòng 2: Ngày sinh
* Dòng 3,4,5: 3 điểm thi tương ứng. Các điểm thi đều đảm bảo hợp lệ (từ 0 đến 10).

**Output:**

In ra danh sách thí sinh đã sắp xếp theo tổng điểm giảm dần. Nếu 2 thí sinh bằng điểm nhau thì thí sinh nào xuất hiện trước sẽ viết trước.  Mỗi thí sinh viết trên một dòng gồm: mã, tên, ngày sinh và tổng điểm. Các thông tin cách nhau đúng 1 khoảng trống.  Điểm tổng được làm tròn đến 1 số sau dấu phẩy.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Kết quả** |
| 3  Nguyen Van A  12/12/1994  3.5  7.0  5.5  Nguyen Van B  1/9/1994  7.5  9.5  9.5  Nguyen Van C  6/7/1994  4.5  4.5  5.0 | 2 Nguyen Van B 1/9/1994 26.5  1 Nguyen Van A 12/12/1994 16.0  3 Nguyen Van C 6/7/1994 14.0 |

#include<stdio.h>

#include<string.h>

struct sv{

int stt;

char ten[1001];

char sinh[1001];

float d1;

float d2;

float d3;

float tong;

};

int main(){

int t,i,j;

scanf("%d",&t);

struct sv a[t+1],x;

float max=0;

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

scanf("\n");

gets(a[i].ten);

scanf("\n");

gets(a[i].sinh);

scanf("\n");

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

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

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

a[i].stt=i;

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

}

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

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

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

x=a[j];

a[j]=a[i];

a[i]=x;

}

}

}

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

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

}

}