Mức độ 1

Mức độ 2

Mức độ 3

Mức độ 4

Mức độ 5

Link youtobe tham khảo https://www.youtube.com/channel/UC5bcTGPVo30W3B5XclfVu0w/videos

Dạng 1. Sinh kế tiếp

1. XÂU NHỊ PHÂN KẾ TIẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01001/top)

Cho xâu nhị phân X[], nhiệm vụ của bạn là hãy đưa ra xâu nhị phân tiếp theo của X[]. Ví dụ X[] =”010101” thì xâu nhị phân tiếp theo của X[] là “010110”.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu nhi phân X.
* T, X[] thỏa mãn ràng buộc: 1≤T≤100; 1≤length(X)≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  010101  111111 | 010110  000000 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int X[1000], n;

void sinh(){

int i=n;

while(i>0&&X[i]==1) i--;

if(i==0)

for(int i=1;i<=n;i++) X[i]=0;

else

for(int j=i;j<=n;j++) X[j]=1-X[j];

return;

}

void in(){

for(int i=1;i<=n;i++) cout<<X[i];

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

n=s.size();

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

X[i]=s[i-1]-'0';

}

sinh();

in();

}

return 0;

}

2. TẬP CON KẾ TIẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01002/top)

Cho hai số N, K và một tập con K phần tử X[] =(X1, X2,.., XK) của 1, 2, .., N. Nhiệm vụ của bạn là hãy đưa ra tập con K phần tử tiếp theo của X[]. Ví dụ N=5, K=3, X[] ={2, 3, 4} thì tập con tiếp theo của X[] là {2, 3, 5}.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là hai số N và K; dòng tiếp theo đưa vào K phần tử của X[] là một tập con K phần tử của 1, 2, .., N.
* T, K, N, X[] thỏa mãn ràng buộc: 1≤T≤100; 1≤K≤N≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int X[1001],n,k;

void sinh(){

int i=k;

while(i>0&&X[i]==n-k+i) i--;

if(i==0){

for(int j=1;j<=k;j++) X[j]=j;

}

else{

X[i]=X[i]+1;

for(int j=i+1;j<=n;j++) X[j]=X[i]+(j-i);

}

return;

}

void in(){

for(int i=1;i<=k;i++) cout<<X[i]<<" ";

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n>>k;

for(int i=1;i<=k;i++) cin>>X[i];

sinh();

in();

}

}

3.  HOÁN VỊ KẾ TIẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01003/top)

Cho số tự nhiên N và một hoán vị X[] của 1, 2, .., N. Nhiệm vụ của bạn là đưa ra hoán vị tiếp theo của X[]. Ví dụ N=5, X[] = {1, 2, 3, 4, 5} thì hoán vị tiếp theo của X[] là {1, 2, 3, 5, 4}.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số N; dòng tiếp theo đưa vào hoán vị X[] của 1, 2, .., N.
* T, N, X[] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

#include<bits/stdc++.h>

using namespace std;

int n,X[1001];

void sinh(){

int i=n-1;

while(i>0&&X[i]>X[i+1]) i--;

if(i==0) for(int j=1;j<=n;j++) X[j]=j;

else{

int k=n;

while(X[k]<X[i]) k--;

swap(X[k],X[i]);

int l=i+1,r=n;

while(l<=r){

swap(X[l],X[r]);

l++,r--;

}

}

return;

}

void in(){

for(int i=1;i<=n;i++) cout<<X[i]<<" ";

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n;

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

sinh();

in();

}

}

4. XÂU AB CÓ ĐỘ DÀI N

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01007/top)

Xâu ký tự str được gọi là xâu AB nếu mỗi ký tự trong xâu hoặc là ký tự ‘A’ hoặc là ký tự ‘B’. Ví dụ xâu str=”ABBABB” là xâu AB độ dài 6. Nhiệm vụ của bạn là hãy liệt kê tất cả các xâu AB có độ dài n.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên n.
* T, n thỏa mãn ràng buộc: 1≤T≤10; 1≤n≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng. Mỗi xâu cách nhau 1 khoảng trống.

|  |  |
| --- | --- |
| Input | Output |
| 2  2  3 | AA AB BA BB  AAA AAB ABA ABB BAA BAB BBA BBB |

//by hung nguyen

#include <iostream>

using namespace std;

int n;

char X[1000];

bool ok;

void sinh(){

int i=n;

while((i>0)&&(X[i]=='B')) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n; j++) X[j]='B'-(X[j]-'A');

return;

}

void in(){

for(int j=1; j<=n; j++) cout << X[j];

cout<<" ";

}

int main() {

int t;cin>>t;

while(t--){

cin>>n;

ok=false;

for(int i=1;i<=n;i++) X[i]='A';

while(!ok){

in();

sinh();

}

cout<<endl;

}

return 0;

}

5.  SINH TỔ HỢP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01004/top)

Cho hai số nguyên dương N và K. Nhiệm vụ của bạn là hãy liệt kê tất cả các tập con K phần tử của 1, 2, .., N. Ví dụ với N=5, K=3 ta có 10 tập con của 1, 2, 3, 4, 5 như sau: {1, 2, 3}, {1, 2, 4},{1, 2, 5},{1, 3, 4},{1, 3, 5},{1, 4, 5},{2, 3, 4},{2, 3, 5},{2, 4, 5},{3, 4, 5}.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một cặp số tự nhiên N, K được viết trên một dòng.
* T, n thỏa mãn ràng buộc: 1≤T≤100; 1≤k ≤ n≤15.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  3  5  3 | 123 124 134 234  123 124 125 134 135 145 234 235 245 345 |

//by hung nguyen

#include <iostream>

using namespace std;

int X[1000],n,k;

bool ok;

void sinh(){

int i=k;

while((i>0)&&(X[i]==n-k+i)) i--;

if(i==0){

ok=true;return;

}

else{

X[i]=X[i]+1;

for(int j=i+1;j<=n;j++) X[j]=X[i]+(j-i);

}

return;

}

void in(){

for(int j=1; j<=k; j++) cout << X[j];

cout<<" ";

}

int main() {

int t;cin>>t;

while(t--){

cin>>n>>k;

ok=false;

for(int i=1;i<=k;i++) X[i]=i;

while(!ok){

in();

sinh();

}

cout<<endl;

}

return 0;

}

6.  SINH HOÁN VỊ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01005/top)

Cho số nguyên dương N. Nhiệm vụ của bạn là hãy liệt kê tất cả các hoán vị của 1, 2, .., N. Ví dụ với N = 3 ta có kết quả: 123, 132, 213, 231, 312, 321.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên N được viết trên một dòng.
* T, n thỏa mãn ràng buộc: 1≤T, N≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2  2  3 | 12  21  123 132  213  231  312  321 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int X[1000],n;

bool ok;

void sinh(){

int i=n-1;

while((i>0)&&(X[i]>X[i+1])) i--;

if(i==0){

ok=true;return;

}

else{

int k=n;

while(X[k]<X[i]) k--;//neu X[k]<X[i] bo qua

swap(X[k],X[i]);

//lat nguoc phan tu ben phai cua X[i]

int l=i+1,r=n;

while(l<=r){

swap(X[l],X[r]);

l++;r--;

}

}

return;

}

void in(){

for(int j=1; j<=n; j++) cout << X[j];

cout<<" ";

}

int main() {

int t;cin>>t;

while(t--){

cin>>n;

ok=false;

for(int i=1;i<=n;i++) X[i]=i;

while(!ok){

in();sinh();

}

cout<<endl;

}

return 0;

}

7.  HOÁN VỊ NGƯỢC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01006/top)

Cho số nguyên dương N. Nhiệm vụ của bạn là hãy liệt kê tất cả các hoán vị của 1, 2, .., N theo thứ tự ngược. Ví dụ với N = 3 ta có kết quả: 321, 312, 231, 213, 132, 123.

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên N được viết trên một dòng.
* T, n thỏa mãn ràng buộc: 1≤T, N≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2  2  3 | 21 12  321 312 231 213 132 123 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int X[1000],n;

bool ok;

void sinh(){

int i=n-1;

while((i>0)&&(X[i]<X[i+1])) i--;

if(i==0){

ok=true;return;

}

else{

int k=n;

while(X[k]>X[i]) k--;//neu X[k]<X[i] bo qua

swap(X[k],X[i]);

//lat nguoc phan tu ben phai cua X[i]

int l=i+1,r=n;

while(l<=r){

swap(X[l],X[r]);

l++;r--;

}

}

return;

}

void in(){

for(int j=1; j<=n; j++) cout << X[j];

cout<<" ";

}

int main() {

int t;cin>>t;

while(t--){

cin>>n;

ok=false;

for(int i=1;i<=n;i++) X[i]=n-i+1;

while(!ok){

in();sinh();

}

cout<<endl;

}

return 0;

}

8. XÂU NHỊ PHÂN CÓ K BIT 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01008/top)

Hãy in ra tất cả các xâu nhị phân độ dài N, có K bit 1 theo thứ tự từ điển tăng dần.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 20). Mỗi test gồm 2 số nguyên N, K (1 ≤ K ≤ N ≤ 16).

**Output:**Với mỗi test, in ra đáp án tìm được, mỗi xâu in ra trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4 2  3 2 | 0011  0101  0110  1001  1010  1100  011  101  110 |

//by hung nguyen

#include <iostream>

using namespace std;

int n,k,X[1000];

bool ok;

void sinh(){

int i=n;

while((i>0)&&(X[i]==1)) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n; j++) X[j]=1-X[j];

return;

}

int test(){

int dem=0;

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

if(X[i]==1) dem++;

}

if(dem==k) return 1;

return 0;

}

void in(){

for(int j=1; j<=n; j++) cout << X[j];

cout<<endl;

}

int main() {

int t;cin>>t;

while(t--){

cin>>n>>k;

ok=false;

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

if(i>n-k) X[i]=1;// toi uu luot sinh

else X[i]=0;

}

while(!ok){

if(test()) in();

sinh();

}

}

return 0;

}

9.  THUẬT TOÁN SINH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_001/top)

Một xâu nhị phân độ dài *n* được gọi là thuận nghịch hay đối xứng nếu đảo ngược xâu nhị phân đó ta vẫn nhận được chính nó. Cho số tự nhiên *n* (n nhập từ bàn phím). Hãy viết chương trình liệt kê tất cả các xâu nhị phân thuận nghịch có độ dài *n*.  Hai phần tử khác nhau của xâu thuận nghịch được ghi cách nhau một vài khoảng trống.

Ví dụ với n = 4 ta tìm được 4 xâu nhị phân thuận nghịch như dưới đây.

                      0                      0                      0                      0

                               0                      1                      1                      0

                               1                      0                      0                      1

                               1                      1                      1                      1

//cach 1 :sinh

#include<bits/stdc++.h>

using namespace std;

int n,X[1001];

bool ok;

void sinh(){

int i=n;

while(i>0&&X[i]==1) i--;

if(i==0){

ok=true;return;

}

else

for(int j=i;j<=n;j++) X[j]=1-X[j];

}

int test(){

for(int i=1;i<=n/2;i++) if(X[i]!=X[n+1-i]) return 0;

return 1;

}

void in(){

for(int i=1;i<=n;i++) cout<<X[i]<<" ";

cout<<endl;

}

int main(){

// int t;cin>>t;

// while(t--){

cin>>n;

for(int i=1;i<=n;i++) X[i]=0;

ok=false;

while(!ok){

if(test()) in();

sinh();

}

//}

}

// cach 2 quay lui

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,x[1000];

void Try(int i){

for(int j=0;j<=1;j++){

x[i]=j;

if(i==n){

int test=1;

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

if(x[i]!=x[n-i+1]){

test=0;break;

}

}

if(test==1){

for(int l=1;l<=n;l++) cout<<x[l]<<" ";

cout<<endl;

}

}

else Try(i+1);

}

}

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

cin>>n;

Try(1);

return 0;

}

10. Tổng dãy con = K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_002/top)

Cho dãy A[] gồm N số tự nhiên khác nhau và số tự nhiên K. Hãy viết chương trình liệt kê tất cả các dãy con của dãy số A[] sao cho tổng các phần tử trong dãy con đó đúng bằng K. Dữ liệu vào trên bàn phím (n=5, K=50), 5 số dòng thứ 2 là các phần tử dãy A:

   5                      50

   5                      10                    15                    20                    25

Các dãy con thoả mãn điều kiện tìm được liệt kê trên màn hình:

* Mỗi dòng ghi lại một dãy con. Hai phần tử khác nhau của dãy con được viết cách nhau bởi một  khoảng trống.
* Dòng cuối cùng ghi lại số các dãy con có tổng các phần tử đúng bằng K tìm được.

           10                    15                    25

           5                      20                    25

          5                      10                    15                    20

           3

Cach 1: sinh

#include<bits/stdc++.h>

using namespace std;

int n,k,X[1001],a[1001],dem;

bool ok;

void sinh(){

int i=n;

while(i>0&&X[i]==1) i--;

if(i==0){

ok=true;return;

}

else

for(int j=i;j<=n;j++) X[j]=1-X[j];

}

int test(){

int sum=0;

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

if(X[i]==1){

sum+=a[i];

if(sum>k) return 0;

}

}

if(sum==k) return 1;

return 0;

}

void in(){

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

if(X[i]==1) cout<<a[i]<<" ";

}

cout<<endl;

}

int main(){

cin>>n>>k;

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

for(int i=1;i<=n;i++) X[i]=0;

ok=false;

while(!ok){

if(test()){

in();dem++;

}

sinh();

}

cout<<dem<<endl;

}

Cach 2: quay lui

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,a[1000],b[1000],dem;

void Try(int i){

for(int j=0;j<=1;j++){

b[i]=j;

if(i==n){

int sum=0;

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

if(b[i]) sum+=a[i];

}

if(sum==k){

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

if(b[i]) cout<<a[i]<<" ";

}

dem++;

cout<<endl;

}

}

else Try(i+1);

}

}

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

cin>>n>>k;

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

dem=0;

Try(1);

cout<<dem<<endl;

return 0;

}

11. HAHAHA

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01019/top)

Khi chat chit vui vẻ, anh em chiến hữu thường hay gõ HAHA để thể hiện sự sảng khoái. Đôi khi cũng có thể gõ HAHAAAAA chẳng hạn cho thêm phần nhấn mạnh.

Giả sử một xâu ký tự được coi là HAHA nếu thỏa mãn các điều kiện:

* Ký tự đầu tiên phải là chữ H, ký tự cuối cùng phải là chữ A
* Không có hai chữ H nào liền nhau

Cho trước độ dài N, hãy liệt kê tất cả các xâu ký tự HAHA theo thứ tự từ điển.

**Input**

* Dòng đầu ghi số bộ test T (không quá 10)
* Mỗi bộ test ghi độ dài N (2 <= N < 16)

**Output**

* Với mỗi bộ test, ghi ra tất cả các xâu HAHA tìm được theo thứ tự từ điển, mỗi xâu viết trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  4 | HA  HAAA  HAHA |

#include<bits/stdc++.h>

using namespace std;

int n;

char X[1001];

bool ok;

void sinh(){

int i=n;

while(i>0&&X[i]=='H') i--;

if(i==0){

ok=true;return;

}

else{

for(int j=i;j<=n;j++) X[j]='H'-(X[j]-'A');

}

}

int test(){

if(X[n]=='H') return 0;

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

if(X[i]==X[i+1]&&X[i]=='H') return 0;

return 1;

}

void in(){

for(int i=1;i<=n;i++) cout<<X[i];

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n;

X[1]='H';

for(int i=2;i<=n;i++) X[i]='A';

ok=false;

while(!ok){

if(test()) in();

sinh();

}

}

}

12. XÂU NHỊ PHÂN TRƯỚC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01020/top)

Cho xâu nhị phân X[], nhiệm vụ của bạn là hãy đưa ra xâu nhị phân trước của X[]. Ví dụ X[] =”111111” thì xâu nhị phân trước của X[] là “111110”. Với xâu X[] =“000001” thì xâu nhị trước của X[] là “000000”. Chú ý: nếu xâu dữ liệu trong input là xâu đầu tiên thì trước nó sẽ là xâu cuối cùng.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu nhi phân X.
* T, X[] thỏa mãn ràng buộc: 1≤T≤100; 1≤length(X)≤103.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  010101  111111 | 010100  111110 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int X[1005],n;

string s;

void sinh(){

int i=n;

while(i>0&&X[i]==0) i--;

if(i==0){

for(int i=1;i<=n;i++) X[i]=1;

}

else{

for(int j=i;j<=n;j++) X[j]=1-X[j];

}

return;

}

void in(){

for(int i=1;i<=n;i++) cout<<X[i];

cout<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>s;

n=s.size();

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

X[i+1]=s[i]-'0';

}

sinh();

in();

}

return 0;

}

13. ĐẶT TÊN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01025/top)

Vương quốc PTIT sử dụng bảng chữ cái gồm N chữ cái Latinh viết hoa. Quy tắc đặt tên của gia đình Hoàng gia  PTIT là chọn ra K chữ cái (không trùng nhau) và sắp xếp lại theo thứ tự từ điển.

Hãy liệt kê tất cả các cái tên có thể có của gia đình Hoàng gia PTIT

**Input**

* Dòng đầu ghi số bộ test T (không quá 10)
* Mỗi bộ test ghi 2 số N và K (3 < K < N < 16)

**Output**

* Với mỗi bộ test, ghi ra tất cả các cái tên có thể được tạo ra, mỗi kết quả viết trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  4 2 | AB  AC  AD  BC  BD  CD |

//by hung nguyen

#include <iostream>

using namespace std;

int n,k;

char X[1000];

bool ok;

void sinh(){

int i=k-1;

while((i>=0)&&(X[i]==(n-k+i)+'A')) i--;

if(i<0){

ok=true;return;

}

else{

X[i]=X[i]+1;

for(int j=i+1;j<n;j++) X[j]=X[i]+(j-i);

}

return;

}

void in(){

for(int j=0; j<k; j++) cout << X[j];

cout<<endl;

}

int main() {

int t;cin>>t;

while(t--){

cin>>n>>k;

ok=false;

for(int i=0;i<k;i++) X[i]=i+'A';

while(!ok){

in();

sinh();

}

}

return 0;

}

14. PHÂN TÍCH SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01016/top)

Cho số nguyên dương N. Nhiệm vụ của bạn là hãy liệt kê tất cả các cách phân tích số tự nhiên N thành tổng các số tự nhiên nhỏ hơn hoặc bằng N. Phép hoán vị vủa một cách được xem là giống nhau. Ví dụ với N = 5 ta có kết quả là: (5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1), (1, 1, 1, 1, 1) .

Input:

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên N được viết trên một dòng.
* T, n thỏa mãn ràng buộc: 1≤T, N≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  5 | (4) (3 1) (2 2) (2 1 1) (1 1 1 1)  (5) (4 1)  (3 2)  (3 1 1) (2 2 1)  (2 1 1 1)  (1 1 1 1 1) |

//by hung nguyen

// code tham khao

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,a[1001];

bool ok;

void init(){

k=1;

a[k]=n;

}

void sinh(){

int i=k;

while(i>0&&a[i]==1) i--;

if(i==0){

ok=true;return;

}

else{

a[i]--;

int d=k-i+1;

k=i;

for(int j=i+1;j<=i+d/a[i];j++)

a[j]=a[i];

k+=d/a[i];

if(d%a[i]) a[++k]=d%a[i];

}

}

void in(){

cout<<"(";

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

cout<<a[i];

if(i!=k) cout<<" ";

}

cout<<") ";

}

int main(){

int t;cin>>t;

while(t--){

cin>>n;

init();

ok=false;

while(!ok){

in();sinh();

}

cout<<endl;

}

}

15. MÃ GRAY 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01012/top)

Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình liệt kê các mã Gray có độ dài n.

Input:

* Dòng đầu tiên là số lượng test T.
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một số tự nhiên n.
* T, n thỏa mãn ràng buộc: 1≤T, n≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  3  4 | 000 001 011 010 110 111 101 100  0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000 |

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

//G1(bit)=01

//G2(bit)=0(G1)+1(~G1)=00 01 11 10

//G3(bit)=0(G2)+1(~G2)=000 001 011 010 110 111 101 100

int n;

void Gray1(){

if (n <= 0) return;//n<=0 return luon

vector<string> a;//luu cac ma

a.push\_back("0");

a.push\_back("1");

int i, j;

for (i = 2; i <=pow(2,n-1);i\*=2){//moi lan lap tao ra 2\*i ma

for (j = i-1 ; j >= 0 ; j--)//gap doi luong ma 1 0-> 10 11 01 00

a.push\_back(a[j]);

for (j = 0 ; j < i ; j++)//noi 0 vao nua dau

a[j] = "0" + a[j];

for (j = i ; j < 2\*i ; j++)//noi 1 vao nua sau

a[j] = "1" + a[j];

}

for (i = 0 ; i < a.size() ; i++ )

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

}

int main(){

int t;cin>>t;

while(t--){

cin>>n;

Gray1 ();

cout<<endl;

}

return 0;

}

16. MÃ GRAY 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01017/top)

Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình chuyển đổi một xâu mã nhị phân X có độ dài n thành một xâu mã Gray.

Input:

* Dòng đầu tiên là số lượng test T.
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một xâu nhị phân độ dài n.
* T, n thỏa mãn ràng buộc: 1≤T, n≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  01001  01101 | 01101  01011 |

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

//01101 giu nguyen 0/0#1->1/1=1->0/1#0->1/0#1->1//01001

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

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

if(i==0) cout<<s[i];

else{

if(s[i]!=s[i-1]) cout<<1;

else cout<<0;

}

}

cout<<endl;

}

return 0;

}  
17. MÃ GRAY 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01013/top)

Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình chuyển đổi một xâu mã Gray X có độ dài n thành một xâu mã nhị phân.

Input::

* Dòng đầu tiên là số lượng test T.
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một xâu mã Gray độ dài n.
* T, n thỏa mãn ràng buộc: 1≤T, n≤10.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  01101  01011 | 01001  01101 |

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

//nguoc bai gray 3

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

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

if(i==0) cout<<s[i];

else{

if(s[i]=='1') s[i]=1-(s[i-1]-'0')+'0';

else s[i]=s[i-1];

cout<<s[i];

}

}

cout<<endl;

}

return 0;

}

18.  XÂU AB ĐẶC BIỆT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01009/top)

Một xâu kí tự S = (s1, s2, .., sn) được gọi là xâu AB độ dài n nếu với mọi siÎS thì si hoặc là kí tự A hoặc si là kí tự B . Ví dụ xâu S = “ABABABAB” là một xâu AB độ dài 8. Cho số tự nhiên N và số tự nhiên K (1£K<N£15 được nhập từ bàn phím), hãy viết chương trình liệt kê tất cả các xâu AB có độ dài N chứa **duy nhất** một dãy K kí tự A liên tiếp.

**Dữ liệu vào** chỉ có một dòng ghi hai số N và K.

**Kết quả** ghi ra màn hình theo khuôn dạng:

* Dòng đầu tiên ghi lại số các xâu AB thỏa mãn yêu cầu bài toán;
* Những dòng kế tiếp, mỗi dòng ghi lại một xâu AB thỏa mãn. Các xâu được ghi ra theo thứ tự từ điển.

**Ví dụ:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 5 3 | 5  AAABA  AAABB  ABAAA  BAAAB  BBAAA |

// cach 1 sinh

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n,k,X[1000];

string s;

bool ok;

vector<string> res;

void sinh(){

int i=n-k;

while((i>0)&&(X[i]==1)) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n-k; j++) X[j]=1-X[j];

return;

}

void test(){

for(int l=1;l<=n-k;l++){

//th1 cong dau chuoi neu dau la B

if(l==1&&X[1]==1){

string tmp=s;

for(int p=1;p<=n-k;p++){

if(X[p]==0) tmp+="A";

else tmp+="B";

}

res.push\_back(tmp);

}

//th2 cong vao cuoi chuoi neu cuoi chuoi la B

if(l==n-k&&X[l]==1){

string tmp="";

for(int p=1;p<=n-k;p++){

if(X[p]==0) tmp+="A";

else tmp+="B";

}

tmp+=s;

res.push\_back(tmp);

}

//th3 co the cong vao giua neu co 2 B lien tiep(B s B)

if(X[l]&&X[l-1]&&l>1){

string tmp="";

for(int p=1;p<=n-k;p++){

if(p==l) tmp+=s;

if(X[p]==0) tmp+="A";

else tmp+="B";

}

res.push\_back(tmp);

}

}

}

int main() {

cin>>n>>k;

for(int i=0;i<k;i++) s+="A";

ok=false;

for(int i=1;i<=n-k;i++) X[i]=0;

while(!ok){

test();

sinh();

}

sort(res.begin(),res.end());

cout<<res.size()<<endl;

for(int i=0;i<res.size();i++) cout<<res[i]<<endl;

return 0;

}

//c2 quay lui

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

//quay lui

int n,k,a[100];

bool ok;

string s;

vector<string> res;

void Try(int i){

for(int j=0;j<=1;j++){

a[i]=j;

if(i==n-k){//xau co n-k ky tu con lai

for(int l=1;l<=n-k;l++){

if(l==1&&a[l]==1){

string tmp=s;

for(int p=1;p<=n-k;p++){

if(a[p]==0) tmp+="A";

else tmp+="B";

}

res.push\_back(tmp);//day tmp vao trong res

}

if(l==n-k&&a[l]==1){

string tmp="";

for(int p=1;p<=n-k;p++){

if(a[p]==0) tmp+="A";

else tmp+="B";

}

tmp+=s;

res.push\_back(tmp);

}

if(a[l]&&a[l-1]&&l>1){

string tmp="";

for(int p=1;p<=n-k;p++){

if(p==l) tmp+=s;

if(a[p]==0) tmp+="A";

else tmp+="B";

}

res.push\_back(tmp);

}

}

}

else Try(i+1);

}

}

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

cin>>n>>k;

for(int i=0;i<k;i++) s+="A";//xau dung xau co k ky tu A

Try(1);

sort(res.begin(),res.end());//sx theo thu tu tu dien

cout<<res.size()<<endl;//sluong test tm

for(int i=0;i<res.size();i++) cout<<res[i]<<endl;

return 0;

}

19. TẬP QUÂN SỰ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01010/top)

Tại Chương Mỹ Resort, vào nửa đêm, cả trung đội nhận lệnh tập trung ở sân. Mỗi chiến sỹ được đánh số từ 1 đến N (1<N<40). Giám thị yêu cầu chọn ra một dãy K chiến sỹ để tập đội ngũ và cứ lần lượt duyệt hết tất cả các khả năng chọn K người như vậy từ nhỏ đến lớn (theo số thứ tự). Bài toán đặt ra là cho một nhóm K chiến sỹ hiện đang phải tập đội ngũ, hãy tính xem trong lượt chọn K người tiếp theo thì mấy người trong nhóm cũ sẽ được tạm nghỉ. Nếu đã là nhóm cuối cùng thì tất cả đều sẽ được nghỉ.

**Dữ liệu vào:**Dòng đầu ghi số bộ test, không quá 20. Mỗi bộ test viết trên hai dòng

* Dòng 1: hai số nguyên dương N và K (K<N)
* Dòng 2 ghi K số thứ tự của các chiến sỹ đang phải tập đội ngũ (viết từ nhỏ đến lớn)

**Kết quả:** Với mỗi bộ dữ liệu in ra số lượng chiến sỹ được tạm nghỉ.

**Ví dụ:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 3  5 3  1 3 5  5 3  1 4 5  6 4  3 4 5 6 | 1  2  4 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,X[1001],ok;

void sinh(){

int i=k;

while(i>0&&X[i]==n-k+i) i--;

if(i==0){

ok=1;return;

}

else{

X[i]=X[i]+1;

for(int j=i+1;j<=n;j++) X[j]=X[i]+(j-i);

}

}

void in(int A[]){

int dem=0;

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

if(A[X[i]]==0) dem++;

cout<<dem;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n>>k;

int A[n]={0};

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

cin>>X[i];

A[X[i]]=1;

}

ok=0;

sinh();

if(ok==1) cout<<k;

else in(A);

cout<<endl;

}

}

20. HOÁN VỊ TIẾP THEO CỦA CHUỖI SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01011/top)

Hãy viết chương trình nhận vào một chuỗi (có thể khá dài) các ký tự số và đưa ra màn hình hoán vị kế tiếp của các ký tự số đó (với ý nghĩa là hoán vị có giá trị lớn hơn tiếp theo nếu ta coi chuỗi đó là một giá trị số nguyên). Chú ý: Các ký tự số trong dãy có thể trùng nhau.

Ví dụ:               *123 -> 132*

*279134399742 -> 279134423799*

Cũng có trường hợp sẽ không thể có hoán vị kế tiếp. Ví dụ như khi đầu vào là chuỗi 987.

**Dữ liệu vào:** Dòng đầu tiên ghi số nguyên  t là số bộ test (1 ≤ t ≤ 1000).  Mỗi bộ test có một dòng, đầu tiên là số thứ tự bộ test, một dấu cách, sau đó là chuỗi các ký tự số, tối đa 80 phần tử.

**Kết quả:**Với mỗi bộ test hãy đưa ra một dòng gồm thứ tự bộ test, một dấu cách, tiếp theo đó là hoán vị kế tiếp hoặc chuỗi “BIGGEST” nếu không có hoán vị kế tiếp.

**Ví dụ:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 3  1 123  2 279134399742  3 987 | 1 132  2 279134423799  3 BIGGEST |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int x,X[1000],n;

void sinh(){

int i=n-1;

while((i>0)&&(X[i]>=X[i+1])) i--;//lon hon or bang co the lui tiep

if(i<=0) cout<<"BIGGEST"<<endl;

else{

int k=n;

while(X[k]<=X[i]) k--;//neu X[k]<=X[i] bo qua

swap(X[k],X[i]);

//lat nguoc phan tu ben phai cua X[i]

int l=i+1,r=n;

while(l<=r){

swap(X[l],X[r]);

l++;r--;

}

for(int j=1; j<=n; j++) cout <<X[j];

cout<<endl;

}

}

int main(){

int t;cin>>t;

string s;

while(t--){

cin>>x>>s;

n=s.length();

cout<<x<<" ";

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

X[i]=int(s[i-1]-'0');//vi s[i] bat dau tu 0

}

sinh();

}

return 0;

}

21. CHỌN SỐ TỪ MA TRẬN VUÔNG CẤP N

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02008/top)

Cho ma trận vuông Ci,j cấp N (*1£ i, j £ N£10*) gồm N2 số tự nhiên và số tự nhiên K (*các số trong ma trận không nhất thiết phải khác nhau và đều không quá 100, K không quá 104*). Hãy viết chương trình lấy mỗi hàng, mỗi cột duy nhất một phần tử sao cho tổng các phần tử này đúng bằng K.

**Dữ liệu vào:** Dòng 1 ghi hai số N và K. N dòng tiếp theo ghi ma trận C.

**Kết quả:** dòng đầu ghi số cách tìm được. Mỗi dòng tiếp theo ghi một cách theo vị trí của số đó trong lần lượt từng hàng của ma trận. Xem ví dụ để hiểu rõ hơn.

**Ví dụ:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 3 10  2 4 3  1 3 6  4 2 4 | 2  1 3 2  3 2 1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//3 10

//

//2 4 3 2

//1 3 6 1 3 2

//4 2 4 3 2 1

//gt tai moi hang chon 1 so sao cho tong tat ca cac so bang k

//in ra tat ca cac cac (vi tri cua cac so tai cac hang)

int n,k,a[100][100],x[100],res;

bool ok[100];

vector<int> v;

void Try(int i){//quay lui sinh hoan vi

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

if(!ok[j]){

x[i]=j;

ok[j]=true;

if(i==n){

res=0;

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

res+=a[h][x[h]];

if(res==k){

for(int l=1;l<=n;l++) v.push\_back(x[l]);//vi phai in ra so cap tm trc nen phai luu tam vao vector

}

}

else Try(i+1);

ok[j]=false;

}

}

}

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

// int t; cin >> t;

// while (t--) {

cin>>n>>k;

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

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

Try(1);

cout<<v.size()/n;//in so luong cac th thoa man

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

if(i%n==0) cout<<endl;//du so phan tu 1 hang thi xd

cout<<v[i]<<" ";

}

return 0;

}

22.  TÌM BỘI SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01015/top)

Cho số nguyên N. Nhiệm vụ của bạn cần tìm số nguyên X nhỏ nhất là bội của N, và X chỉ chứa hai chữ số 0 và 9.

**Input:** Dòng đầu tiên là số lượng bộ test T (T ≤ 10000). Mỗi bộ test chứa số nguyên N trên một dòng (1 ≤ N ≤ 500).

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2  5  11 | 90  90  99 |

//by hung nguyen

#include <iostream>

#include <vector>

#include <queue>

#include <string>

using namespace std;

vector<string> v;

void sang() {

queue<string> q;

q.push("9");

int count = 5000;

while (count--) {

string str1 = q.front();

q.pop();

v.push\_back(str1);

string str2 = str1;

q.push(str1.append("0"));

q.push(str2.append("9"));

}

}

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

sang();

int t; cin >> t;

while (t--) {

int n; cin >> n;

int i;

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

if (stoll(v[i]) % n == 0) break;

}

cout << v[i] << endl;

}

}

23. TẬP HỢP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01014/top)

Xét tất cả các tập hợp các số nguyên dương có các phần tử khác nhau và không lớn hơn số n cho trước. Nhiệm vụ của bạn là hãy đếm xem có tất cả bao nhiêu tập hợp có số lượng phần tử bằng k và tổng của tất cả các phần tử trong tập hợp bằng s?

Các tập hợp là hoán vị của nhau chỉ được tính là một.

Ví dụ với n = 9, k = 3, s = 23, {6, 8, 9} là tập hợp duy nhất thỏa mãn.

**Input:**Gồm nhiều bộ test (không quá 100 test).

Mỗi bộ test gồm 3 số nguyên n, k, s với 1 ≤ n ≤ 20, 1 ≤ k ≤ 10 và 1 ≤ s ≤ 155. Input kết thúc bởi 3 số 0.

**Output:**Với mỗi test in ra số lượng các tập hợp thỏa mãn điều kiện đề bài.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9 3 23  9 3 22  10 3 28  16 10 107  20 8 102  20 10 105  20 10 155  3 4 3  4 2 11  0 0 0 | 1  2  0  20  1542  5448  1  0  0 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//quay lui tim tong

//duyet tu j=a[i-1]+1->j<=n-k+1

//vd k=3, a[]max nhat =7 cac gtri con lai min nhat=1;

//vd 9 3 23

//thuc hien quay lui 1 2 3/ 1 2 4/1 2 5/1 2 6/1 3 4/1 3 5/......

int n,k,s,dem,a[11];

void Try(int i){

for(int j=a[i-1]+1;j<=n-k+i;j++){

a[i]=j;

if(i==k){

int sum=0;

for(int h=1;h<=k;h++) sum+=a[h];

if(sum==s) dem++;

}

else Try(i+1);

}

return;

}

int main(){

while(1){

cin>>n>>k>>s;

//dk dung

if(n==0&&k==0&&s==0) break;

dem=0;

// khoi tao gt ban dau cho mang

for(int i=0;i<=10;i++) a[i]=0;

Try(1);

cout<<dem<<endl;

}

return 0;

}

24. PHƯƠNG ÁN TỐI ƯU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_003/top)

Cho *a*i, *c*i , W*,* *N* (i =1, 2,..,N; N100) là những số nguyên dương  và tập hợp

Hãy viết chương trình tìm phương án tối ưu *XOPT =(x1,x2,..,xN)* và giá trị tối ưu *FOPT=F(XOPT)*của hàm mục tiêu

**;**với  .

      Dữ liệu vào cho bởi file data.in theo khuôn dạng sau:

* Dòng đầu tiên ghi lại số tự nhiên *N* và W. Hai số được viết cách nhau một vài khoảng trống;
* Dòng kế tiếp ghi lại *N* số *c*j (*j=1,2,..,N*). Hai số được viết cách nhau một vài khoảng trống;
* Dòng kế tiếp ghi lại *N* số *a*j (*j=1,2,..,N*). Hai số được viết cách nhau một vài khoảng trống;

Giá trị tối ưu *FOPT* và phương án *XOPT* tìm được sẽ liệt kê trên màn hình theo khuôn dạng:

* Dòng đầu tiên ghi lại giá trị tối ưu *FOPT*;
* Dòng kế tiếp ghi lại phương án tối ưu *XOPT*. Hai phần tử khác nhau của phương án tối ưu được viết cách nhau bởi một khoảng trống.

Ví dụ sau sẽ minh họa khuôn dạng cho file data.in  của bài toán.

**data.in**

4                      10

6          5          3          7

5          4          6          5

#include<bits/stdc++.h>

using namespace std;

int n,w,a[101],c[101];

bool ok;

void data(){

cin>>n>>w;

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

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

}

void sinh(int b[]){

// Cach 1

// int i=n;

// while(i>0){

// if(b[i]==1) b[i]=0;

// else{

// b[i]=1;break;

// }

// i--;

// }

// if(i==0) ok=false;

// cach2 sinh thuong

int i=n;

while(i>0&&b[i]==1) i--;

if(i==0){

ok=false;return;

}

else

for(int j=i;j<=n;j++) b[j]=1-b[j];

}

int main(){

data();

int b[101]={0},d[101]={0},FOPT=0;// khoi tao gia tri toi uu

ok=true;

while(ok){

int XOPT=0,tmp=0;//khoi tao phong an toi uu va gia tri tam thoi cua gia tri toi uu

sinh(b);

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

tmp+=a[i]\*b[i];

XOPT+=c[i]\*b[i];

if(XOPT>w) break;

}

if(XOPT<=w&&tmp>FOPT){

FOPT=tmp;// cap nhap gia tri toi uu

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

d[i]=b[i];// cap nhap phuong an toi uu

}

}

cout<<FOPT<<endl;

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

cout<<d[i]<<" ";

}

25. DÃY CON TĂNG DẦN BẬC K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_004/top)

Cho dãy gồm *N* số phân biệt *A*N = {*a1, a2, .., aN* } và số tự nhiên*K* (*KN<=100*). Ta gọi một *dãy con tăng dần bậc K* của dãy số *A*N là một dãy các số gồm *K* phần tử trong dãy đó thỏa mãn tính chất tăng dần. Bài toán được đặt ra là in ra màn hình  số các *dãy con tăng dần bậc K* của dãy số *A*N. Ví dụ :

**Input:**5    3

2    5    15   10    20

* Dòng đầu tiên ghi lại hai số *N* và *K* tương ứng với số phần tử của dãy số và bậc của dãy con.
* Dòng kế tiếp : *N* số của dãy số *A*N, các số trong dãy không lớn hơn 100.

thì Output : 7 (số các dãy con tăng dần bậc *3* của dãy số *A*N)

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

bool stop;

int a[1001],b[1001],n,k,res;

bool test[1001];

void Try(int i){

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

if(!test[j] && j > b[i-1]){

b[i] = j;

test[j] = 1;

if(i == k){

int ans = 0;

for (int h=1;h<k;h++){

if(a[b[h]] > a[b[h+1]])

ans = 1;

}

if(ans== 0)

res++;

}

else

Try(i + 1);

test[j] = 0;

}

}

}

int main(){

res=0;

cin>>n>>k;

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

Try(1);

cout << res << endl;

return 0;

}

26. TẬP CON LIỀN KỀ PHÍA TRƯỚC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA01018/top)

Cho hai số N, K và một tập con K phần tử X[] =(X1, X2,.., XK) của 1, 2, .., N. Nhiệm vụ của bạn là hãy đưa ra tập con K phần tử trước đó của X[]. Ví dụ N=5, K=3, X[] ={2, 3, 5} thì tập con trước đó của X[] là {2, 3, 4}. Chú ý nếu tập con trong input là đầu tiên thì trước đó là tập con cuối cùng.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là hai số N và K; dòng tiếp theo đưa vào K phần tử của X[] là một tập con K phần tử của 1, 2, .., N.
* T, K, N, X[] thỏa mãn ràng buộc: 1≤T≤100; 1≤K≤N≤103.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//gt code bai co 3 th

//so dung sau=so dung truoc+1

//th1 neu i==1&&a[i]=1 thi ta dua ve cau hinh cuoi cung vd 123-> 345

//th2 (con lai) tai vi tri do a[i]=a[i]-1;cac vi tri sau do lay max vd 1 2 4 5->1 2 3 5

int n,k,a[1001];

void in(){

for(int i=1;i<=k;i++) cout<<a[i]<<" ";

cout<<endl;

}

void sinh(){

int i=k;

while(i>1&&a[i]==(a[i-1]+1)) i--;

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

for(int j=1;j<=k;j++) a[j]=n-k+j;

}

else{

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

for(int j=i+1;j<=k;j++) a[j]=n-k+j;

}

return;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>k;

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

sinh();

in();

}

return 0;

}

Dạng 2:Quay lui – Nhánh cận

1. DÃY SỐ 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02001/top)

Cho dãy số A[] gồm n số nguyên dương. Tam giác đặc biệt của dãy số A[] là tam giác được tạo ra bởi n hàng, trong đó hàng thứ 1 là dãy số A[], hàng i là tổng hai phần tử liên tiếp của hàng i-1 (2≤i≤n). Ví dụ A[] = {1, 2, 3, 4, 5}, khi đó tam giác được tạo nên như dưới đây:

[1, 2, 3, 4, 5 ]

[3, 5, 7, 9 ]

[8, 12, 16]

[20, 28]

[48]

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N là số lượng phần tử của dãy số A[]; dòng tiếp theo đưa vào N số của mảng A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤N, A[i] ≤10;

**Output:**

* Đưa ra tam giác tổng của mỗi test theo từng dòng. Mỗi dòng của tam giác tổng được bao bởi ký tự [, ].

|  |  |
| --- | --- |
| Input | Output |
| 1  5  1 2 3 4 5 | [1 2 3 4 5]  [3 5 7 9]  [8 12 16]  [20 28]  [48] |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;cin>>t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

int x=n;

while(x>0){

cout<<"[";

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

if(x==n) cout<<a[i];

else{

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

cout<<a[i];

}

if(i!=x-1) cout<<" ";

}

cout<<"]";

x--;

cout<<endl;

}

}

return 0;

}

2. DÃY SỐ 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02002/top)

Cho dãy số A[] gồm n số nguyên dương. Tam giác đặc biệt của dãy số A[] là tam giác được tạo ra bởi n hàng, trong đó hàng thứ n là dãy số A[], hàng i là tổng hai phần tử liên tiếp của hàng i+1 (1≤i≤n-1). Ví dụ A[] = {1, 2, 3, 4, 5}, khi đó tam giác được tạo nên như dưới đây:

[48]

[20, 28]

[8, 12, 16]

[3, 5, 7, 9 ]

[1, 2, 3, 4, 5 ]

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N là số lượng phần tử của dãy số A[]; dòng tiếp theo đưa vào N số của mảng A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤N, A[i] ≤10;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Mỗi dòng của tam giác tổng được bao bởi ký tự [, ].

|  |  |
| --- | --- |
| Input | Output |
| 1  5  1 2 3 4 5 | [48] [20 28] [8 12 16] [3 5 7 9 ] [1 2 3 4 5 ] |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

void Solve(){

int n;cin>>n;

int a[n+1][n+1];

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

cin>>a[1][j];

}

int cot=n-1;

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

for(int j=1;j<=cot;j++){

a[i][j]=a[i-1][j]+a[i-1][j+1];

}

cot--;

}

cot=n;

vector<vector<int> > res;

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

vector<int> x;

for(int j=1;j<=cot;j++)

x.push\_back(a[i][j]);

cot--;

res.push\_back(x);

}

for(int i=res.size()-1;i>=0;i--){

for(int j=0;j<res[i].size();j++){

if(j==0) cout<<"[";

cout<<res[i][j];

if(j==res[i].size()-1) cout<<"] ";

if(j<res[i].size()-1) cout<<" ";

}

}

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

Solve();

}

}

3. HOÁN VỊ XÂU KÝ TỰ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02005/top)

Cho xâu ký tự S bao gồm các ký tự in hoa khác nhau. Hãy đưa ra tất cả các hoán vị của xâu ký tự S. Ví dụ S=”ABC” ta có kết quả {ABC ACB BAC BCA CAB CBA}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test là một xâu ký tự S được viết trên 1 dòng.
* T, S thỏa mãn ràng buộc: 1≤T≤10; 1≤length(S) ≤10;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2 AB  ABC | AB BA  ABC ACB BAC BCA CAB CBA |

#include<bits/stdc++.h>

using namespace std;

int n;

char X[100];

bool ok;

void sinh(){

int i=n-1;

while((i>0)&&(X[i]-'A'>X[i+1]-'A')) i--;

if(i==0){

ok=true;return;

}

else{

int k=n;

while(X[k]-'A'<X[i]-'A') k--;//neu X[k]<X[i] bo qua

swap(X[k],X[i]);

//lat nguoc phan tu ben phai cua X[i]

int l=i+1,r=n;

while(l<=r){

swap(X[l],X[r]);

l++;r--;

}

}

return;

}

void in(){

for(int j=1; j<=n; j++) cout << X[j];

cout<<" ";

}

int main() {

int t;cin>>t;

while(t--){

string s;cin>>s;

ok=false;

n=s.length();

for(int i=1;i<=n;i++) X[i]='A'+(i-1);

while(!ok){

in();

sinh();

}

cout<<endl;

}

return 0;

}

4.  DI CHUYỂN TRONG MÊ CUNG 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02003/top)

Cho một mê cung bao gồm các khối được biểu diễn như một ma trận nhị phân A[N][N]. Một con chuột đi từ ô đầu tiên góc trái (A[0][0]) đến ô cuối cùng góc phải (A[N-1][N-1]) theo nguyên tắc:

* Down (D): Chuột được phép xuống dưới nếu ô dưới nó có giá trị 1.
* Right (R): Chuột được phép sang phải dưới nếu ô bên phải nó có giá trị 1.

Hãy đưa ra một hành trình của con chuột trên mê cung. Đưa ra -1 nếu chuột không thể đi đến đích.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N là kích cỡ của mê cung; dòng tiếp theo đưa vào ma trận nhị phân A[N][N].
* T, N, A[i][j] thỏa mãn ràng buộc: 1≤T ≤10; 2≤N≤10; 0≤A[i][j] ≤1.

**Output:**

* Đưa ra tất cả đường đi của con chuột trong mê cung theo thứ tự từ điển. Đưa ra -1 nếu chuột không đi được đến đích.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  1  0  0  0  1  1  0  1  0  1  0  0  1  1  1  1  5  1 0 0 0 0  1 1 1 1 1  1 1 0 0 1  0 1 1 1 1  0 0 0 1 1 | DRDDRR  DDRDRRDR DDRDRRRD DRDDRRDR DRDDRRRD DRRRRDDD |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[100][100];

vector<string> v;

bool ok;

void Try(int i,int j,string s){

if(i==1&&j==1&&a[i][j]==0){//neu tai vi tri dau tien ma a[i][j]=0 tra ve false luon

ok=false;return;

}

if(i==n&&j==n&&a[n][n]){//neu co the di den cuoi , ktra xem vitri cuoi co phai =1 ko

v.push\_back(s);//luu vao vector

ok=true;

return;

}

if(i<n&&a[i+1][j]){//di chuyen xuong duoi giu nguyen cot ,tang hang

Try(i+1,j,s+"D");//tang hang,them D

}

if(j<n&&a[i][j+1]){//di chuyen sang phai giu nguyen hang, tang cot

Try(i,j+1,s+"R");//tang cot,cong them R

}

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n;

ok=false;

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

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

Try(1,1,"");//quay lui voi dk ban dau tai vi tri (1,1) va chuoi rong

if(!ok) cout<<"-1";//ko co duong di nao

else{

sort(v.begin(),v.end());//sap xep theo thu tu tu dien

for(int i=0;i<v.size();i++) cout<<v[i]<<" ";

v.clear();//xoa du lieu test truoc

}

cout<<endl;

}

return 0;

}

5.  DÃY CON TỔNG BẰNG K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02006/top)

Cho dãy số A[] = (a1, a2, .., an) và số tự nhiên K. Hãy đưa ra tất cả các dãy con của dãy số A[] sao cho tổng các phần tử của dãy con đó đúng bằng K. Các phần tử của dãy số A[] được giả thuyết là nguyên dương và không có các phần tử giống nhau. Ví dụ với dãy con A[] = {5, 10, 15, 20, 25}, K = 50 ta có 3 dãy con {5, 10, 15, 20}, {5, 20, 25}, {10, 15, 25}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N là số lượng phần tử của dãy số A[] và số K; dòng tiếp theo đưa vào N phần tử của dãy số A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T ≤100; 1≤N≤10; 1≤ K, A[i] ≤100.

**Output:**

* Đưa ra tất cả các dãy con của dãy số A[] thỏa mãn yêu cầu bài toán theo thứ tự từ điển, trong đó mỗi dãy con được bao bởi các ký tự [, ]. Nếu không có dãy con nào thỏa mãn yêu cầu bài toán, hãy đưa ra -1.

|  |  |
| --- | --- |
| Input | Output |
| 2 5 50  5  10 15 20  25 8  53 15  22  14  26  32  9  16  8 | [5 10 15 20] [5 20 25] [10 15 25] [8 9 14 22] [8 14 15 16] [15 16 22] |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k;

int X[100],a[100];//X[i] de luu vi tri cua a[i]

vector<vector<int>> v;//coi nhu mang hai chieu

void Try(int i){//quay lui nhi phan

for(int j=0;j<=1;j++){

X[i]=j;

if(i==n){

int s=0;

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

if(X[h]) s+=a[h];//tai cac vi tri X[l]=1 cong vao s

}

if(s==k){

vector<int> res;

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

if(X[h]) res.push\_back(a[h]);//day a[h] vao vector

v.push\_back(res);//day res vao vector

}

}

else Try(i+1);

}

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n>>k;

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

cin>>a[i];

sort(a+1,a+n+1);

Try(1);

if(v.size()==0) cout<<"-1";//ko co tap hop nao tm

else{

sort(v.begin(),v.end());//sap xep theo tu dien

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

cout<<"[";

for(int j=0;j<v[i].size()-1;j++)

cout<<v[i][j]<<" ";

cout<<v[i][v[i].size()-1]<<"] ";

}

}

v.clear();

cout<<endl;

}

return 0;

}

6. TẬP CON BẰNG NHAU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05009/top)

Cho tập các số A[] = (a1, a2, .., an). Hãy kiểm tra xem ta có thể chia tập A[] thành hai tập con sao cho tổng các phần tử của hai tập con bằng nhau hay không. Đưa ra YES nếu có thể thực hiện được, ngược lại đưa ra NO.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N là số lượng phần tử của dãy số A[]; dòng tiếp theo đưa vào N phần tử của dãy số A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T ≤100; 1≤N≤100; 1≤ A[i] ≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2 4 1 5 11 5 3 1 3 5 | YES NO |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//quy hoach dong

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

int n;cin>>n;

int a[n],sum=0;

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

cin>>a[i];sum+=a[i];

}

if(sum%2==1) cout<<"NO";

else{

int s=sum/2;

int F[s+1]={0};

F[0]=1;

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

for(int j=s;j>=a[i];j--){

if(F[j-a[i]]==1) F[j]=1;

//j=a[i]+(j-a[i])

}

}

if(F[s]==1) cout<<"YES";

else cout<<"NO";

}

cout<<endl;

}

return 0;

}

7.  DI CHUYỂN TRONG MA TRẬN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02012/top)

Cho ma trận A[M][N]. Nhiệm vụ của bạn là đếm tất cả các đường đi từ phần tử A[0][0] đến phần tử A[M-1][N-1]. Bạn chỉ được phép dịch chuyển xuống dưới hoặc sang phải phần tử liền kề với vị trí hiện tại.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số M, N tương ứng với số hàng và số cột của ma trận; dòng tiếp theo đưa vào các phần tử của ma trận A[][]; các số được viết cách nhau một vài khoảng trống.
* T, M, N, A[i][j] thỏa mãn ràng buộc: 1≤T ≤10; 1≤M, N, A[i][j]≤100.

**Output:**

* Đưa ra số cách di chuyển của mỗi test theo từng dòng.
* Giải thích test 1: Có 3 cách di chuyển là [1 4 5 6], [1 2 5 6] và [1 2 3 6].

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,m,a[101][101],dem;

void Try(int i,int j){

if(i==n&&j==m){

dem++;

return;

}

if(i<n){

Try(i+1,j);

}

if(j<m){

Try(i,j+1);

}

//if((i<n&&j<m&&!a[i+1][j]&&a[i][j+1])||i>n||j>m) return;

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n>>m;

dem=0;

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

for(int j=1;j<=m;j++) cin>>a[i][j];

if(n==1&&m==1) cout<<"0"<<endl;

else{

Try(1,1);

cout<<dem<<endl;

}

}

return 0;

}

8. SẮP XẾP QUÂN HẬU 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02016/top)

Cho một bàn cờ vua có kích thước n \* n, ta biết ràng quân hậu có thể di chuyển theo chiều ngang, dọc, chéo. Vấn đề đặt ra rằng, có n quân hậu, bạn cần đếm số cách đặt n quân hậu này lên bàn cờ sao cho với 2 quân hậu bất kì, chúng không “ăn” nhau.

**Input:**Dòng đầu ghi số bộ test T (T<5). Mỗi bộ test ghi một số nguyên dương n duy nhất (không quá 10)

**Output:**Ghi kết quả mỗi bộ test trên một dòng. Số cách đặt quân hậu.

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,dem;

bool hang[11],cot[11],Xuoi[22],Nguoc[22];

void Try(int i){

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

if(!hang[j]&&!cot[j]&!Nguoc[i+j-1]&&!Xuoi[i-j+n]){//neu cac vi tri do chua dc danh dau

hang[j]=cot[j]=Nguoc[i+j-1]=Xuoi[i-j+n]=true;//danh dau la da di

if(i==n) dem++;

else Try(i+1);//neu chua dem diem cuoi thi tiep tuc quay lui

hang[j]=cot[j]=Nguoc[i+j-1]=Xuoi[i-j+n]=false;

}

}

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n;

dem=0;

Try(1);

cout<<dem<<endl;

}

return 0;

}

9.0. MÃ SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02031/top)

Số lượng máy tính ở các phòng thực hành nhà A3 tăng lên nhanh chóng. Để gán mã cho các máy tính của PTIT người ta sử dụng mã gồm 2\*N ký tự, trong đó:

* N ký tự đầu tiên là hoán vị của N chữ cái in hoa đầu tiên, tính từ A.
* N ký tự tiếp theo là các ký tự số bất kỳ từ 1 đến N (có thể trùng nhau).

Người ta ước tính chỉ cần N = 5 là đủ để gán mã cho toàn bộ máy tính kể cả khi mở rộng quy mô các phòng thực hành. Hãy viết chương trình liệt kê các mã tạo được với giá trị N cho trước.

**Input**

Chỉ có duy nhất số N (1 < N < 6)

**Output**

Ghi ra lần lượt các mã khác nhau tạo được theo thứ tự từ điển, mỗi mã ghi trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | AB11  AB12  AB21  AB22  BA11  BA12  BA21  BA22 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[10];

char X[10];

bool ok;

void sinh(){

int i=n-2;

while((i>=0)&&(X[i]>X[i+1])) i--;

if(i<0){

ok=true;return;

}

else{

int k=n-1;

while(X[k]<X[i]) k--;

swap(X[k],X[i]);

int l=i+1,r=n-1;

while(l<=r){

swap(X[l],X[r]);

l++;r--;

}

}

return;

}

void in(){

for(int i=0;i<n;i++) cout<<X[i];

for(int i=1;i<=n;i++) cout<<a[i];

cout<<endl;

}

void slove(){

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

a[h]=1;

}

in();

for(int i=n;i>=0;i--){

if(i==0) return;

if(a[i]<n){

a[i]++;

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

a[j]=1;

}

}

in();

for(int k=n;k>=0;k--){

if(a[k]<n){

i=k+1;

break;

}

}

}

}

int main() {

cin>>n;

ok=false;

for(int i=0;i<n;i++) X[i]=i+'A';

while(!ok){

slove();

sinh();

}

return 0;

}

9.DI CHUYỂN TRONG MÊ CUNG 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02004/top)

Cho một mê cung bao gồm các khối được biểu diễn như một ma trận nhị phân A[N][N]. Một con chuột đi từ ô đầu tiên góc trái (A[0][0]) đến ô cuối cùng góc phải (A[N-1][N-1]) theo nguyên tắc:

* Down (D): Chuột được phép xuống dưới nếu ô dưới nó có giá trị 1.
* Right (R): Chuột được phép sang phải dưới nếu ô bên phải nó có giá trị 1.
* Left (L): Chuột được phép sang trái dưới nếu ô bên trái nó có giá trị 1.
* Up (U): Chuột được phép lên trên nếu ô trên nó có giá trị 1.

Hãy đưa ra tất cả các hành trình của con chuột trên mê cung. Đưa ra -1 nếu chuột không thể đi đến đích.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N là kích cỡ của mê cung; dòng tiếp theo đưa vào ma trận nhị phân A[N][N].
* T, N, A[i][j] thỏa mãn ràng buộc: 1≤T ≤10; 2≤N≤8; 0≤A[i][j] ≤1.

**Output:**

* Đưa ra các xâu ký tự được sắp xếp, trong đó mỗi xâu là một đường đi của con chuột trong mê cung. In ra đáp án theo thứ tự từ điển. Đưa ra -1 nếu chuột không đi được đến đích.

|  |  |
| --- | --- |
| Input | Output |
| 3 4 1 0 0 0  1 1 0 1  0 1 0 0  0 1 1 1 4 1 0 0 0  1 1 0 1  1 1 0 0  0 1 1 1  5  1 0 0 0 0  1 1 1 1 1  1 1 1 0 1  0 0 0 0 1  0 0 0 0 1 | DRDDRR  DDRDRR DRDDRR  DDRRURRDDD DDRURRRDDD DRDRURRDDD DRRRRDDD |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[100][100];

bool ok[100][100];

vector<string>v;

// quay lui

void Try(int i,int j,string s){

//xet o dau va o cuoi matran

if(a[1][1]==0&&a[n][n]==0) return;

//neu co the di den o cuoi cung -> luu vao vector

if(i==n&&j==n) v.push\_back(s);

// di chuyen xuong duoi

if(a[i+1][j]==1&&i<n&&!ok[i+1][j]){//!ok[i+1][j]:o nay chua tung dc di den trc do

ok[i][j]=true;

Try(i+1,j,s+"D");

ok[i][j]=false;

}

//di chuyen sang phai

if(a[i][j+1]==1&&j<n&&!ok[i][j+1]){

ok[i][j]=true;

Try(i,j+1,s+"R");

ok[i][j]=false;

}

//di chuyen len tren

if(a[i-1][j]==1&&i>1&&!ok[i-1][j]){

ok[i][j]=true;

Try(i-1,j,s+"U");

ok[i][j]=false;

}

//di chuyen sang trai

if(a[i][j-1]==1&&j>1&&!ok[i][j-1]){

ok[i][j]=true;

Try(i,j-1,s+"L");

ok[i][j]=false;

}

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

//xoa du lieu cua v

v.clear();

cin>>n;

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

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

cin>>a[i][j];

//danh dau tat ca bang false(chua xet)

ok[i][j]=false;

}

}

Try(1,1,"");

if(v.size()==0) cout<<"-1"<<endl;

else{

//sap xep theo thu tu tu dien

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

//in

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

cout<<v[i]<<" ";

}

cout<<endl;

}

}

return 0;

}

10.ĐỔI CHỖ CÁC CHỮ SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02007/top)

Cho số tự nhiên K và xâu ký tự các chữ số S.  Nhiệm vụ của bạn là đưa ra số lớn nhất bằng cách thực hiện nhiều nhất K lần đổi chỗ các ký tự trong S. Ví dụ K =3 và S = “1234567” ta được “7654321”.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số K; dòng tiếp theo là xâu ký tự S.
* T, K, S thỏa mãn ràng buộc: 1≤T ≤100; 1≤K≤10; 1≤.lenght(S)≤7.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 3 4 1234567 3 3435335 2 1034 | 7654321 5543333 4301 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int k;

string s;

//y tuong bai toan

//vd 4 1 5 2 2 5 -> 5 5 1 2 2 4

//chu ko phai 5 5 4 2 2 1

//nghia la 4 se doi cho voi max xa nhat (5 o cuoi) chu ko phai 5 o giua

//nen ta phai xuyet lon nhat so voi i tu cuoi ve

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>k>>s;

int n=s.size();

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

char max=s[n-1];

int vitrimax=n-1;

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

if(max<s[j]){//so sanh ky tu

max=s[j];

vitrimax=j;

}

}

if(max>s[i]&&k>0){

swap(s[i],s[vitrimax]);

k--;

}

}

cout<<s<<endl;

}

return 0;

}

11. Số nguyên Tố

Cho ba số N, P, S. Trong đó, P là một số nguyên tố. Nhiệm vụ của bạn là đưa ra tất cả N số nguyên tố tính từ P có tổng bằng S. Ví dụ với S = 28, P=7, N =2 ta có kết quả 11 + 17 = 28. Với N = 3, P = 2, S = 23 ta có kết quả : {3, 7, 13}, {5, 7, 11}

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ ba số S, P, N được viết trên một dòng.
* S, P, N thỏa mãn ràng buộc: 1≤T ≤100; 1 ≤ N ≤ 10; 2≤S, P≤200.

**Output:**

* Với mỗi test, dòng đầu tiên in ra số lượng đáp án tìm được. Mỗi dòng tiếp theo in ra kết quả tìm được theo thứ tự từ điển.

|  |  |
| --- | --- |
| Input | Output |
| 2 2  7  28 3  2  23 | 1  11 17  2  3 7 13  5 7 11 |

// by hung nguyen

#include<bits/stdc++.h>

#include<cstring>

using namespace std;

int n, p, s;

vector< vector<int> >vtr;

int sum;

bool nt(int i){

if(i<2) return false;

else if(i<=3) return true;

else if(i%2==0||i%3==0) return false;

else{

for(int j=5; j<=sqrt(i); j+=6)

if(i%j==0||i%(j+2)==0) return false;

return true;

}

}

void Try(int n, int p, int s, vector<int>res){

if(sum>s) return;

if(res.size()==n&&sum==s){

vtr.push\_back(res);

return;

}

for(int i=p+1; i<=200; i++){

if(nt(i)){

if(sum+i<=s&&res.size()<n){

sum+=i;

res.push\_back(i);

Try(n, i, s, res);

res.pop\_back();

sum-=i;

}

}

}

}

int main(){

int t;

cin>>t;

while(t--){

cin>>n>>p>>s;

sum=0;

vtr.clear();

vector<int>res;

Try(n, p, s, res);

sort(vtr.begin(), vtr.end());

cout<<vtr.size()<<endl;

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

for(int j=0; j<vtr[i].size(); j++) cout<<vtr[i][j]<<" ";

cout<<endl;

}

}

return 0;

}

12. SẮP XẾP QUÂN HẬU 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02017/top)

Cho một bàn cờ 8 x 8, mỗi ô có một giá trị A[i][j] nhất định (0 ≤ A[i][j] ≤ 100), tương ứng với điểm số đạt được nếu như bạn đặt một quân cờ vào đó.

Nhiệm vụ của bạn là đặt 8 quân hậu lên bàn cờ, sao cho không có 2 quân nào ăn nhau, và số điểm đạt được là lớn nhất.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 8 dòng, mỗi dòng 8 số nguyên mô tả bàn cờ.

**Output:**Với mỗi test, in ra đáp án trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  1 2 3 4 5 6 7 8  9 10 11 12 13 14 15 16  17 18 19 20 21 22 23 24  25 26 27 28 29 30 31 32  33 34 35 36 37 38 39 40  41 42 43 44 45 46 47 48  48 50 51 52 53 54 55 56  57 58 59 60 61 62 63 64 | 260 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,dem,b[11],a[10][10];

bool hang[11],cot[11],Xuoi[22],Nguoc[22];

void Try(int i){

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

//xet duong di tai cac diem chua di

if(!hang[j]&&!cot[j]&&!Nguoc[i+j-1]&&!Xuoi[i-j+n]){

b[i]=j;

//danh dau la da di qua

hang[j]=cot[j]=Nguoc[i+j-1]=Xuoi[i-j+n]=true;

if(i==n){

int res=0;

//cong tong cac diem thoa man

for(int l=1;l<=n;l++) res+=a[l][b[l]];

dem=max(res,dem);

}

else Try(i+1);

//danh dau lai la chua xet

hang[j]=cot[j]=Nguoc[i+j-1]=Xuoi[i-j+n]=false;

}

}

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

n=8;

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

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

memset(hang,false,sizeof(hang));

memset(cot,false,sizeof(cot));

memset(Nguoc,false,sizeof(Nguoc));

memset(Xuoi,false,sizeof(Xuoi));

dem=0;

Try(1);

cout<<dem<<endl;

}

return 0;

}

13. PHÂN TÍCH SỐ 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02039/top)

Cho số nguyên dương N. Nhiệm vụ của bạn là hãy liệt kê tất cả các cách phân tích số tự nhiên N thành tổng các số tự nhiên nhỏ hơn hoặc bằng N. Phép hoán vị của một cách được xem là giống nhau. Ví dụ với N = 5 ta có kết quả là: (5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1), (1, 1, 1, 1, 1) .

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên N được viết trên một dòng.
* T, n thỏa mãn ràng buộc: 1≤T, N≤10.

**Output:**

* Dòng đầu tiên là số lượng cách phân tích thỏa mãn. Dòng tiếp theo liệt kê đáp án theo mẫu ví dụ đã cho.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  5 | 5  (4) (3 1) (2 2) (2 1 1) (1 1 1 1)  7  (5) (4 1) (3 2) (3 1 1) (2 2 1) (2 1 1 1) (1 1 1 1 1) |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[100],k=1;

bool ok;

void init(){

k=1;

a[k]=n;

}

void sinh(){

int i=k;

while(i>=1&&a[i]==1) i--;

if(i==0){

ok=true;return;

}

else{

a[i]--;

int d=k-i+1;

int p=d/a[i],r=d%a[i];

k=i;

if(p!=0){

for(int j=1;j<=p;j++){

a[i+j]=a[i];

}

k+=p;

}

if(r!=0){

a[k+1]=r;

k++;

}

}

return;

}

void in(string &s){

if(n==10&&k==1){

s+='(';s+='1';s+='0';s+=')';s+=' ';

}

else{

s+='(';

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

s+=(a[i]+’0’;

if(i!=k) s+=' ';

}

s+=')';s+=' ';

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

ok=false;

init();

int dem=0;

string s="";

while(!ok){

in(s);

sinh();

dem++;

}

cout<<dem<<endl;

cout<<s<<endl;

}

return 0;

}

14. TỔ HỢP SỐ CÓ TỔNG BẰNG X

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02032/top)

Cho mảng A[] gồm N số nguyên dương phân biệt và số X. Nhiệm vụ của bạn là tìm phép tổ hợp các số trong mảng A[] có tổng bằng X. Các số trong mảng A[] có thể được sử dụng nhiều lần. Mỗi tổ hợp các số của mảng A[] được in ra theo thứ tự không giảm các số. Ví dụ với A[] = {2, 4, 6, 8}, X = 8 ta có các tổ hợp các số như sau:

{2, 2, 2, 2}, {2, 2, 4}, {2, 6}, {4, 4}, {8}.

**Input:**Dòng đầu tiên đưa vào số lượng bộ test T. Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số N và X; dòng tiếp theo đưa vào N số của mmảng A[]; các số được viết cách nhau một vài khoảng trống. T, N, X, A[i] thỏa mãn ràng buộc: 1≤T ≤10; 1≤X, A[i]≤100. N ≤ 20.

**Output:**Đưa ra kết quả mỗi test theo từng dòng. Đầu tiên là số lượng  tổ hợp thỏa mãn. Mỗi tổ hợp được bao bởi cặp ký tự { } và cách nhau một dấu cách. Đưa ra -1 nếu không có tổ hợp nào thỏa mãn yêu cầu bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4  8  2  4  6  8  2 9  10 11 | 5 {2 2 2 2} {2 2 4} {2 6} {4 4} {8}  -1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,X[100];

vector<vector<int>> res;

void Try(int i,int sum,vector<int> ans){

if(sum>k) return;

else if(sum==k){

res.push\_back(ans);return;

}

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

if(sum+X[j]<=k){

ans.push\_back(X[j]);

Try(j,sum+X[j],ans);

ans.pop\_back();//xoa phan tu cuoi cung de co cong tiep

}

}

}

void in(){

if(res.size()==0){

cout<<"-1"<<endl;return;

}

cout<<res.size()<<" ";

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

int x=res[i].size();

cout<<"{";

for(int j=0;j<x-1;j++){

cout<<res[i][j]<<" ";

}

cout<<res[i][x-1]<<"} ";

}

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n>>k;

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

cin>>X[i];

}

sort(X+1,X+n+1);

vector<int> ans;

Try(1,0,ans);

in();

res.clear(),ans.clear();

}

}

15. TẬP CON CỦA XÂU KÝ TỰ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02045/top)

Cho một xâu ký tự S không có ký tự lặp lại. Hãy đưa ra tất cả các tập con của xâu ký tự S theo thứ tự từ điển.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu ký tự.
* T, S thỏa mãn ràng buộc: 1≤T≤100; 1≤length(S)≤16.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 3 abc | a ab abc ac b bc c |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[20];

string s;

vector<string> res;

bool ok;

void sinh(){

int i=n-1;

while(i>=0&&a[i]==1)i--;

if(i<0){

ok=true;return ;

}

for(int j=i;j<=n;j++) a[j]=1-a[j];

}

int main() {

int t; cin >> t;

while (t--) {

res.clear();

cin >> n;

cin >> s;

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

ok=false;

while(!ok){

sinh();

string tmp="";

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

if (a[i] == 1){

tmp += s[i];

}

}

if(ok==false)

res.push\_back(tmp);

}

sort(res.begin(), res.end());

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

cout << res[i] << " ";

}

cout << endl;

}

return 0;

}

18. BIẾN ĐỔI VỀ 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02041/top)

Cho số nguyên dương N. Hãy đếm số bước ít nhất để đưa N về 1 bằng cách thực hiện ba thao tác dưới đây:

* Nếu N chia hết cho 2 bạn có thể giảm N = N/2.
* Nếu N chia hết cho 3 bạn có thể giảm N = N/3.
* Giảm N đi 1.

**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 N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤100000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 10 6 | 3  2 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[100001];

void init(){

a[1] = 0; a[2] = 1; a[3] = 1;

for (int i = 4; i <= 100000; i++) {

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

if (i % 3 == 0) {

a[i] = min(a[i], a[i / 3] + 1);

}

if (i % 2 == 0) {

a[i] = min(a[i], a[i / 2] + 1);

}

}

}

int main() {

int t; cin >> t;

init();

while (t--) {

cin >> n;

cout << a[n] << endl;

}

return 0;

}

19. MÁY ATM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02011/top)

Một máy ATM hiện có n (n ≤ 30) tờ tiền có giá trị t[1], t[2], …, t[n]. Hãy tìm cách trả ít tờ nhất với số tiền đúng bằng S (các tờ tiền có giá trị bất kỳ và có thể bằng nhau, mỗi tờ tiền chỉ được dùng một lần).

**Input:**Dòng đầu tiên ghi số bộ test T (T<10). Mỗi bộ test gồm 2 số nguyên n và S (S ≤ 109). Dòng thứ hai chứa n số nguyên t[1], t[2], …, t[n] (t[i] ≤ 109)

**Output:**Với mỗi bộ test ghi ra số tờ tiền ít nhất phải trả.

     Nếu không thể tìm được kết quả, in ra -1.

**Ví dụ**

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

 #include<bits/stdc++.h>

using namespace std;

int n,s,a[50],X[50];

bool ok;

void sinh(int k){

int i=k;

while(i>0&&X[i] == n-k+i) i--;

if(i<=0){

ok=true;return;

}

else{

X[i]++;

for(int j=i+1;j<=k;j++) X[j]=X[j-1]+1;

}

return;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>s;

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

int ok2=0;

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

ok=false;

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

X[k]=k;//luu vi tri to tien thu j;

while(!ok){

int tmp=0;

for(int j=1;j<=i;j++) tmp+=a[X[j]];

if(tmp==s){

cout<<i<<endl;ok2=1;break;

}

sinh(i);

}

if(ok2==1) break;

}

if(ok2==0) cout<<-1<<endl;

}

return 0;

}

20. CHIA MẢNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02009/top)

Cho mảng các số nguyên A[] gồm N phần tử. Hãy chia mảng số nguyên A[] thành K tập con khác rỗng sao cho tổng các phần tử của mỗi tập con đều bằng nhau. Mỗi phần tử thuộc tập con xuất hiện duy nhất một lần trong tất cả các tập con. Ví dụ với A[] = {2, 1, 4, 5, 6}, K =3 ta có kết quả {2, 4}, {1, 5}, {6}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số N và K; dòng tiếp theo đưa vào N số của mmảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T ≤100; 1≤N, K≤20, 0≤A[i]≤100.

**Output:**

* Đưa ra 1 nếu có thể chia tập con thành K tập thỏa mãn yêu cầu bài toán, ngược lại đưa ra 0.

|  |  |
| --- | --- |
| Input | Output |
| 2 5 3 2 1 4 5 6 5 3 2 1 5 5 6 | 1 0 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,k;

cin>>n>>k;

int a[n];

int max=0,sum=0;

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

cin>>a[i];

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

sum+=a[i];

}

if(sum%k==0&&max<=sum/k) cout<<"1"<<endl;

else cout<<"0"<<endl;

}

return 0;

}

21. TỔ HỢP SỐ CÓ TỔNG BẰNG X

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02010/top)

Cho mảng A[] gồm N số nguyên dương phân biệt và số X. Nhiệm vụ của bạn là tìm phép tổ hợp các số trong mảng A[] có tổng bằng X. Các số trong mảng A[] có thể được sử dụng nhiều lần. Mỗi tổ hợp các số của mảng A[] được in ra theo thứ tự không giảm các số. Ví dụ với A[] = {2, 4, 6, 8}, X = 8 ta có các tổ hợp các số như sau:

[2, 2, 2, 2], [2, 2, 4], [2, 6], [4, 4], [8].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số N và X; dòng tiếp theo đưa vào N số của mmảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T ≤10; 1≤X, A[i]≤100. N ≤ 20.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Mỗi đường tổ hợp được bao bởi cặp ký tự [, ]. Đưa ra -1 nếu không có tổ hợp nào thỏa mãn yêu cầu bài toán.

|  |  |
| --- | --- |
| Input | Output |
| 1 4  8  2  4  6  8 | [2 2 2 2] [2 2 4] [2 6] [4 4] [8] |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,X[100];

vector<vector<int>> res;

void Try(int i,int sum,vector<int> ans){

if(sum>k) return;

else if(sum==k){

res.push\_back(ans);return;

}

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

if(sum+X[j]<=k){

ans.push\_back(X[j]);

Try(j,sum+X[j],ans);

ans.pop\_back();//xoa phan tu cuoi cung de co cong tiep

}

}

}

// test cham ko co cach giua [][]

void in(){

if(res.size()==0){

cout<<"-1"<<endl;return;

}

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

int x=res[i].size();

cout<<"[";

for(int j=0;j<x-1;j++){

cout<<res[i][j]<<" ";

}

cout<<res[i][x-1]<<"]";

}

cout<<endl;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n>>k;

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

cin>>X[i];

}

sort(X+1,X+n+1);

vector<int> ans;

Try(1,0,ans);

in();

res.clear(),ans.clear();

}

}

23. NGƯỜI DU LỊCH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02022/top)

Cho n thành phố đánh số từ 1 đến n và các tuyến đường giao thông hai chiều giữa chúng, mạng lưới giao thông này được cho bởi mảng C[1…n, 1…n] ở đây C[i][j] = C[j][i] là chi phí đi đoạn đường trực tiếp từ thành phố i đến thành phố j.

Một người du lịch xuất phát từ thành phố 1, muốn đi thăm tất cả các thành phố còn lại mỗi thành phố đúng 1 lần và cuối cùng quay lại thành phố 1. Hãy chỉ ra chi phí ít nhất mà người đó phải bỏ ra.

**Dữ liệu vào:** Dòng đầu tiên là số nguyên n – số thành phố (n ≤ 15); n dòng sau, mỗi dòng chứa n số nguyên thể hiện cho mảng 2 chiều C.

**Kết quả:** Chi phí mà người đó phải bỏ ra.

**Ví dụ:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 4  0 20 35 10  20 0 90 50  35 90 0 12  10 50 12 0 | 117 |

// by hung nguyen

// nguoi di du lich

#include<bits/stdc++.h>

using namespace std;

vector<vector<int>> A;

int ans = INT\_MAX, n, min\_dis = INT\_MAX,tmp=0;

bool v[100] = {};

void backTrack(int x, int count){

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

if(v[i]) continue;

tmp+= A[x][i];

v[i] = true;

if(count == n){

ans = min(ans, tmp+ A[i][1]);

}

else if(tmp+ min\_dis < ans && count < n) backTrack(i, count + 1);

v[i] = false;

tmp -= A[x][i];

}

}

void slove(){

cin >> n;

A.resize(n + 1, vector<int>(n + 1));

for(int i=1; i<=n; i++) for(int j=1; j<=n; j++) cin >> A[i][j], min\_dis = min(min\_dis, A[i][j]);

v[1] = true;

backTrack(1, 2);

cout << ans << endl;

}

int main(){

slove();

return 0;

}

25. TỪ ĐIỂN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02014/top)

Cho tập từ ghi trong trừ điển dic[] và một bảng hai chiều A[M][N] các ký tự. Hãy tạo nên tất cả các từ có mặt trong từ điển dic[] bằng cách nối các ký tự kề nhau trong mảng A[][]. Chú ý, phép nối các ký tự kề nhau trong mảng A[][] được thực hiện theo 8 hướng nhưng không có phần tử A[i][j] nào được lặp lại. Ví dụ với từ điển dic[] ={ “GEEKS”, “FOR”, “QIUZ”, “GO”} và mảng A[][] dưới đây sẽ cho ta kết quả: “GEEKS”, “QUIZ”

|  |  |  |
| --- | --- | --- |
| G | I | Z |
| U | E | K |
| Q | S | E |

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào ba số K, M, N tương ứng với số từ của từ điển dic[], số hàng và số cột của ma trận ký tự A[M][N]; dòng tiếp theo đưa vào K từ của từ điển dic[]; dòng cuối cùng đưa vào các phần tử A[i][j].
* T, K, M, N thỏa mãn ràng buộc: 1≤T ≤10; 1≤K≤100; 1≤ M, N ≤3.

**Output:**

* Đưa ra theo thứ tự tăng dần các từ có mặt trong từ điển dic[] được tạo ra từ ma trận A[][]. Đưa ra -1 nếu không thể tạo ra từ nào thuộc dic[] từ A[][].

|  |  |
| --- | --- |
| Input | Output |
| 1 4  3  3 GEEKS FOR QUIZ GO G I Z  U E K  Q S E | GEEKS QUIZ |

// by hung nguyen

// tu dien

#include<bits/stdc++.h>

using namespace std;

vector<vector<char>> A(10, vector<char>(10, '\0'));

vector<vector<bool>> tham;

map<string, bool> res;

bool ok = false;

int k,n,m;

string s;

void backTrack(int x, int y){

if(res[s]){

ok = true;

cout <<s<< " ";

}

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

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

if(tham[i][j] || A[i][j] == '\0') continue;

tham[i][j] = true;

s+=A[i][j];

backTrack(i, j);

tham[i][j] = false;

s.erase(s.length() - 1);

}

}

}

void slove(){

ok = false;

cin >> k >> n >> m;

A.assign(n + 5, vector<char>(m + 5, '\0'));

vector<string> v(k);

for(int i=0; i<k; i++) cin >> v[i], res[v[i]] = true;

for(int i=1; i<=n; i++) for(int j=1; j<=m; j++) cin >> A[i][j];

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

for(int j=1; j<=m; j++){

//khoi tao lai gia tri

tham.assign(n + 5, vector<bool>(m + 5, false));

s = "";

s+= A[i][j];

tham[i][j] = true;

backTrack(i, j);

}

}

if(!ok) cout << -1;

cout << endl;

}

int main(){

int t;cin >> t;

while(t--){

slove();

res.clear();

}

return 0;

}

27.  ĐƯỜNG ĐI DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02019/top)

Cho đồ thị vô hướng có N đỉnh và M cạnh. Bạn hãy tìm đường đi dài nhất trên đồ thị, sao cho mỗi cạnh chỉ được đi qua nhiều nhất 1 lần.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 10). Mỗi test bắt đầu bằng số nguyên N và M (1 ≤ N, M ≤ 20). Các đỉnh đánh dấu từ 0, 1, …, N-1. M dòng tiếp theo, mỗi dòng gồm 2 số u, v cho biết có cạnh nối giữa uàv.

**Output:**Với mỗi test, in ra đáp án tìm được, mỗi xâu in ra trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3 2  0 1  1 2  15 16  0 2  1 2  2 3  3 4  3 5  4 6  5 7  6 8  7 8  7 9  8 10  9 11  10 12  11 12  10 13  12 14 | 2  12 |

// by hung nguyen

// duong di dai nhat

#include<bits/stdc++.h>

using namespace std;

vector<vector<int>> res;

vector<vector<bool>> test;

int ans,n,m;

void backTrack(int u, int v){

ans = max(ans,v);

for(auto x: res[u]){

if(!test[u][x]){

test[x][u]=test[u][x] = true;

backTrack(x, v + 1);

test[u][x] =test[x][u]= false;

}

}

}

void slove(){

ans = 0;

cin >> n >> m;

res.resize(n+1);

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

int x, y;cin >> x >> y;

res[x].push\_back(y);

res[y].push\_back(x);

}

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

test.assign(n + 1, vector<bool>(n + 1, false));

backTrack(i, 1);

}

cout << ans - 1 << endl;

}

int main(){

int t;cin >>t;

while(t--){

res.clear();

slove();

}

return 0;

}

28. SỐ NHỎ NHẤT CÓ N ƯỚC SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA02020/top)

Cho số nguyên dương N. Nhiệm vụ của bạn là tìm số K nhỏ nhất, sao cho K có đúng N ước. Input đảm bảo rằng đáp án không vượt quá 1018.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤  20).

Mỗi test gồm 1 số nguyên N ( 1 ≤  N ≤  1000).

**Output:**Với mỗi test, in ra đáp án trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  6 | 6  12 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[]={2,3,5,7,11,13,17,19,23,29,31};

long long res;

void Try(int i,long long tmp,long long uoc){

if(uoc>n) return;

else if(uoc==n){

res=min(res,tmp);

}

for(int j=1;;j++){

if(tmp\*a[i]>res) break;

tmp\*=a[i];

Try(i+1,tmp,uoc\*(j+1));

}

}

int main(){

int t;cin >>t;

while(t--){

cin>>n;

res=1e18;

Try(0,1,1);

cout<<res<<endl;

}

return 0;

}

Dạng 3:GIẢI THUẬT THAM LAM

1. ĐỔI TIỀN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03001/top)

Tại ngân hàng có các mệnh giá bằng 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000. Tổng số tiền cần đổi có giá trị bằng N.  Hãy xác định xem có ít nhất bao nhiêu tờ tiền sau khi đổi tiền?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 50).  Mỗi test gồm 1 số nguyên N ( 1 ≤ N ≤ 100 000).

**Output:**Với mỗi test, in ra đáp án trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  70  121 | 2  3 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int N;cin>>N;

int a[10]={1,2,5,10,20,50,100,200,500,1000},dem=0;

while(N>0){

for(int i=9;i>=0;i--){

if(N>=a[i]){

while(N>=a[i]){

dem+=N/a[i];

N=N%a[i];

}

}

}

cout<<dem<<endl;

}

}

return 0;

}

2. NHẦM CHỮ SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03002/top)

Trong một buổi học toán, giáo viên viết 2 số nguyên, A và B, và yêu cầu Tèo thực hiện phép cộng. Tèo không bao giờ tính toán sai, nhưng thỉnh thoảng cậu ta chép các con số một cách không chính xác. Lỗi duy nhất của là ghi nhầm '5' thành '6' hoặc ngược lại.  Cho hai số, A và B, tính tổng nhỏ nhất và lớn nhất mà Tèo có thể nhận được.

**Input:** Có một dòng chứa hai số nguyên dương A và B ( 1 ≤ A, B ≤ 1 000 000).

**Output:** In ra 2 số nguyên cách nhau một dấu cách, tổng nhỏ nhất và lớn nhất có thể nhận được.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Test 1** | **Test 2** | **Test 3** |
| Input: 11 25  Ouput: 36 37 | Input: 1430 4862  Ouput: 6282 6292 | Input: 16796 58786  Ouput: 74580 85582 |

#include<bits/stdc++.h>

using namespace std;

int a,b,r,dem,x,y;

void min(){

x=a,y=b;

dem=0;

while(x!=0){

r=x%10;

if(r==6) a=a-pow(10,dem);

dem++;x/=10;

}

dem=0;

while(y!=0){

r=y%10;

if(r==6) b=b-pow(10,dem);

dem++;y/=10;

}

cout<<a+b<<" ";

}

void max(){

x=a,y=b;

dem=0;

while(x!=0){

r=x%10;

if(r==5) a=a+pow(10,dem);//6-5=1

dem++;x/=10;

}

dem=0;

while(y!=0){

r=y%10;

if(r==5) b=b+pow(10,dem);

dem++;y/=10;

}

cout<<a+b;

}

int main(){

cin>>a>>b;

min();max();

return 0;

}

3.  TÌM MAX

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03003/top)

Cho mảng A[] gồm N phần tử.Nhiệm vụ của bạn là tìm  bằng cách sắp đặt lại các phần tử trong mảng. Chú ý, kết quả của bài toán có thể rất lớn vì vậy bạn hãy đưa ra kết quả lấy modulo với 109+7.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N, A[i] ≤107.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 5 5 3 2 4 1 3 1 2 3 | 40 8 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long MOD=1e9+7;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++)cin>>a[i];

sort(a,a+n);

long long tong=0;

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

tong=((tong%MOD)+(a[i]\*i)%MOD)%MOD;

}

cout<<tong<<endl;

}

return 0;

}

4. TỔNG NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03004/top)

Cho mảng A[] gồm các số từ 0 đến 9. Nhiệm vụ của bạn là tìm tổng nhỏ nhất của hai số được tạo bởi các số trong mảng A[]. Chú ý, tất cả các số trong mảng A[] đều được sử dụng để tạo nên hai số.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤20; 0≤A[i]≤9.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví  dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 6 6 8 4 5 2 3 5 5 3 0 7 4 | 604 82 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long MOD=1e9+7;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++)cin>>a[i];

sort(a,a+n);

long long tong1=0,tong2=0;

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

if(i%2==0) tong1=tong1\*10+a[i];

else tong2=tong2\*10+a[i];

}

cout<<tong1+tong2<<endl;

}

return 0;

}

5.  CHIA MẢNG THÀNH HAI MẢNG CON CÓ TỔNG LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03005/top)

Cho mảng A[] gồm N số nguyên không âm và số K. Nhiệm vụ của bạn là hãy chia mảng A[] thành hai mảng con có kích cỡ K và N-K sao cho hiệu giữa tổng hai mảng con là lớn nhất. Ví dụ với mảng A[] = {8, 4, 5, 2, 10}, K=2 ta có kết quả là 17 vì mảng A[] được chia thành hai mảng {4, 2} và { 8, 5,10} có hiệu của hai mảng con là 23-6=17 là lớn nhất.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N và số K; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ K<N ≤50; 0≤A[i]≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 5 2 8 4 5 2 10 8 3 1 1 1 1 1 1 1 1 | 17 2 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,k;cin>>n>>k;

int a[n];

for(int i=0;i<n;i++)cin>>a[i];

sort(a,a+n);

long long tong=0;

int x=min(k,n-k);

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

if(i<x) tong-=a[i];

else tong+=a[i];

}

cout<<tong<<endl;

}

return 0;

}

6.  SẮP XẾP THAM LAM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03006/top)

Cho mảng A[] gồm N số và thực hiện các thao tác theo nguyên tắc dưới đây:

* Ta chọn một mảng con sao cho phần tử ở giữa của mảng con cũng là phần tử ở giữa của mảng A[] (trong trường hợp N lẻ).
* Đảo ngược mảng con đã chọn trong mảng A[]. Ta được phép chọn mảng con và phép đảo ngược mảng con bao nhiêu lần tùy ý.

Ví dụ với mảng A[] = {1, 6, **3, 4, 5**, 2, 7} ta có câu trả lời là Yes vì: ta chọn mảng con {3, 4, 5} và đảo ngược để nhận được mảng A[]={1, 6, **5, 4, 3**, 2, 7}, chọn tiếp mảng con {6, 5, 4, 3, 2} và đảo ngược ta nhận được mảng A[]={1, 2, 3, 4, 5, 6, 7}. Hãy cho biết ta có thể sắp xếp được mảng A[] bằng cách thực hiện các thao tác kể trên 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 gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N ≤50; 0≤A[i]≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 7 1 6 3 4 5 2 7 7 1 6 3 4 5 7 2 | Yes No |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n],ok=1;

for(int i=0;i<n;i++) cin>>a[i];

if(n==1||n==2) cout<<"Yes";

else{

int x=n/2;

if(x%2==1) if((a[n/2]<a[n/2+1]&&a[n/2]<a[n/2-1])||(a[n/2]>a[n/2+1]&&a[n/2]>a[n/2-1])){

cout<<"No";ok=0;//vd 435 or342

}

if(ok==1){

for(int i=n/2-1;i>0;i--){

if((a[i]>=a[i-1]&&a[n-i-1]<=a[n-i])||(a[i]>=a[n-i]&&a[n-i-1]<=a[i-1])){//vd 13456 or 63451

ok=1;

}

else {

cout<<"No";ok=0;break;

}

}

if(ok==1) cout<<"Yes";

}

}

cout<<endl;

}

return 0;

}

7.  NỐI DÂY 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03010/top)

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số lượng sợi dây N; dòng tiếp theo đưa vào N số A[i] là độ dài của các sợi dây; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤106; 0≤A[i]≤106.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4 4 3 2 6 5 4 2 7 6 9 | 29 62 |

//by hung nguyen

#include <bits/stdc++.h>

using namespace std;

//hang doi uu tien

//chu y neu 2 4 6 7 9 2+4=6 6=6 nen bo qua 1 buoc

// 2+4=6/6+6=12/7+9=16/12+16=28

long long n,a[100001];

long long min(){

priority\_queue<long long, vector<long long>, greater<long long> > pq(a, a + n);//tao hang doi

long long res = 0;//bien kq=0

while (pq.size() > 1) {//neu kich thuoc hang doi lon hon 1

long long first = pq.top();//lay ra 2 soi day ngan nhat tu pq(hang doi)

pq.pop();

long long second = pq.top();

pq.pop();

//cout<<first<<" "<<second<<endl;

res += first + second;//cong cac day do

pq.push(first + second);//chen day moi vao pq

}

return res;

}

int main(){

int t;cin>>t;

while(t--){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

cout<<min()<<endl;

}

return 0;

}

8. SẮP ĐẶT XÂU KÝ TỰ 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03012/top)

Cho xâu ký tự S bao gồm các ký tự in thường. Nhiệm vụ của bạn là kiểm tra xem ta có thể sắp đặt lại các ký tự trong S để hai ký tự giống nhau đều không kề nhau hay không? Đưa ra 1 nếu có thể sắp đặt lại các ký tự trong S thỏa mãn yêu cầu bài toán, ngược lại đưa ra -1.

**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 xâu ký tự S được viết trên một dòng.
* T, S thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 3 geeksforgeeks bbbabaaacd bbbbb | 1 1  -1 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

vector<int> dem(26,0);//coi nhu mang gan tat ca bang 0

int n=s.length(),ok=0;

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

dem[s[i]-'a']++;

}

sort(dem.begin(),dem.end());

if(n%2==1){

if(dem[26-1]<=n/2+1){

cout<<"1";ok=1;

}

}

else {

if(dem[26-1]<=n/2){

cout<<"1";ok=1;

}

}

if(ok==0) cout<<"-1";

cout<<endl;

}

return 0;

}

9. SẮP ĐẶT XÂU KÝ TỰ 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03013/top)

Cho xâu ký tự S bao gồm các ký tự in thường và số D. Nhiệm vụ của bạn là kiểm tra xem ta có thể sắp đặt lại các ký tự trong S để tất cả các ký tự giống nhau đều có khoảng cách là D hay không? Đưa ra 1 nếu có thể sắp đặt lại các ký tự trong S thỏa mãn yêu cầu bài toán, ngược lại đưa ra -1.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số D; dòng tiếp theo là xâu S.
* T, S, D thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000; 1≤D≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2  ABB 2 AAA | 1 -1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//tuong tu xap xep xau ky tu 1(bai xap xep xau ky tu 1 thi k=1)

//gt code vd aaaabbccdd

//ta tim dc max =4(a xuat hien 4 lan)

//4 so a tao ra 3 khoang trong(max-1)a

//moi khoang trong chua k-1 so vi khoang cac bang k

//nen s.size()-max (so cac so con lai bbccdd) >(max-1)\*(k-1) (so khoang trong)

//a b b a c c a d d a

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

string s;int k;

cin>>k>>s;

int dem[260]={0};//tong co 256 ky tu

int max=0;

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

dem[s[i]]++;//dem so lan xuat hien cua ky tu

if(max<dem[s[i]]) max=dem[s[i]];

}

if((max-1)\*(k-1)>s.size()-max) cout<<"-1";

else cout<<1;

cout<<endl;

}

return 0;

}

10. MUA LƯƠNG THỰC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03015/top)

Giả sử bạn là một người nghèo trong địa phương của bạn. Địa phương của bạn có duy nhất một cửa hàng bán lương thực. Cửa hàng của bạn mở cửa tất cả các ngày trong tuần ngoại trừ chủ nhật. Cho bộ ba số N, S, M thỏa mãn ràng buộc sau:

* N : số đơn vị lương thực nhiều nhất bạn có thể mua trong ngày.
* S : số lượng ngày bạn cần được sử dụng lương thực để tồn tại.
* M : số đơn vị lương thực cần có mỗi ngày để bạn tồn tại.

Giả sử bạn đang ở ngày thứ 2 trong tuần và cần tồn tại trong S ngày tới. Hãy cho biết số lượng ngày ít nhất bạn cần phải mua lương thực từ của hàng để tồn tại hoặc bạn sẽ bị chết đói trong S ngày tới.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ 3 số N, S, M được viết trên một dòng.
* T, N, S, M thỏa mãn ràng buộc: 1≤T≤100;  1≤N, S, M ≤30.

**Output:**

* Đưa ra số ngày ít nhất bạn có thể mua lương thực để tồn tại hoặc đưa ra -1 nếu bạn bị chết đói.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 16 10 2 20 10 30 | 2 -1 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int a,b,c;cin>>a>>b>>c;

int x=c\*b,ok=0;

for(int i=1;i<=b-b/7;i++){

if(a\*i>=x){

cout<<i;ok=1;break;

}

}

if(ok==0) cout<<"-1";

cout<<endl;

}

return 0;

}

11. DÃY CON CHUNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03034/top)

Cho dãy số A[], B[] và C[] là dãy không giảm và có lần lượt N, M, K phần tử. Nhiệm vụ của bạn là hãy tìm các phần tử chung của 3 dãy số này.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

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

Dòng tiếp theo gồm N số nguyên A[i], rồi M số nguyên B[i] và K số nguyên C[i].

(0 ≤ A[i], B[i], C[i] ≤ 10^9).

**Output:**

Với mỗi test, in ra trên một dòng là đáp án thu được. Nếu không tìm được đáp án, in ra “NO”.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 3  6 5 8  1 5 10 20 40 80  5 7 20 80 100  3 4 15 20 30 70 80 120  3 5 4  1 5 5  3 4 5 5 10  5 5 10 20  3 3 3  1 2 3  4 5 6  7 8 9 | 20 80  5 5  NO |

// by hung nguyen 2

#include <bits/stdc++.h>

using namespace std;

void in(int n1,int n2,int n3,int a[],int b[],int c[]){

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

bool ok=0;

while(i<n1&&j<n2&&k<n3){

if(a[i]==b[j]&&b[j]==c[k]){

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

i++;j++,k++;

ok=1;

}

else if(a[i]<b[j]) i++;

else if(b[j]<c[k]) j++;

else k++;

}

if(!ok) cout<<"NO";

cout<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

int n1,n2,n3;cin>>n1>>n2>>n3;

int a[n1],b[n2],c[n3];

for(int i=0;i<n1;i++) cin>>a[i];

for(int i=0;i<n2;i++) cin>>b[i];

for(int i=0;i<n3;i++) cin>>c[i];

in(n1,n2,n3,a,b,c);

}

return 0;

}

13. TÍCH LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03022/top)

Cho dãy số A gồm N phần tử là các số nguyên. Hãy tính tích lớn nhất của **2 hoặc 3** phần tử trong dãy.

**Input**

Dòng đầu tiên ghi số N (3 ≤ N ≤ 10000)

Dòng thứ 2 ghi N số của dãy A (|Ai| ≤ 1000)

**Outpput**

Ghi ra kết quả trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  5 10 -2 3 5 2 | 250 |

// by hung nguyen 2

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;cin>>n;

int a[n+5],b[10];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

b[0]=a[0]\*a[1]\*a[2];

b[1]=a[0]\*a[1];

b[2]=a[n-1]\*a[n-2]\*a[n-3];

b[3]=a[n-1]\*a[n-2];

b[4]=a[0]\*a[1]\*a[n-1];

b[5]=a[0]\*a[n-1]\*a[n-2];

sort(b,b+6);

cout<<b[5];

return 0;

}

Bỏ ( XÂU NHỊ PHÂN XEN KẼ)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03036/top)

Một xâu nhị phân được gọi là xen kẽ nếu giá trị 0 ở ngay bên cạnh giá trị 1 và không có hai giá trị nào bằng nhau ở cạnh nhau. Hãy viết chương trình liệt kê các xâu nhị phân xen kẽ có độ dài N.

**Input**

Chỉ có một dòng ghi số N (2 ≤ N ≤ 1000)

**Output**

Ghi ra các xâu nhị phân xen kẽ, mỗi xâu trên một dòng.

**Ví dụ**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;cin>>n;

int a[n];

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

if(i%2==0) a[i]=0;

else a[i]=1;

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

}

cout<<endl;

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

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

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

}

cout<<endl;

return 0;

}

14. GIÁ TRỊ NHỎ NHẤT CỦA BIỂU THỨC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03007/top)

Cho mảng A[], B[] đều có N phần tử. Nhiệm vụ của bạn là tìm giá trị nhỏ nhất của  biểu thức P = A[0]\*B[0] + A[1]\*B[1] + ..+A[N-1]\*B[N-1] bằng cách tráo đổi vị trí các phần tử của cả mảng A[] và B[].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i]; dòng cuối cùng đưa vào N số B[i] các số được viết cách nhau một vài khoảng trống.
* T, N, A[i], B[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N ≤107; 0≤A[i], B[i] ≤1018.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 7 1 6 3 4 5 2 7  1 1 1 2 3 4 3 7 1 6 3 5 5 2 2  0 1 9 0 1 2 3 | 45  27 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long n,sum;

//mang 1 sap xep tang dan nhan mang 2 xap xep giam dan tao ra mang moi

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n;

//vi n<=10^7 nen ko the dung mang thuong-> dung vector

vector<long long > a(n);

vector<long long> b(n);

sum=0;

for(int i=0;i<n;i++) cin>>a[i];

for(int i=0;i<n;i++) cin>>b[i];

//mang a tang dan

sort(a.begin(),a.end());

//mang b giam dan

sort(b.begin(),b.end(),greater<long long>());

//tinh tong

for(int i=0;i<n;i++) sum+=a[i]\*b[i];

cout<<sum<<endl;

}

return 0;

}

15.  SẮP XẾP CÔNG VIỆC 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03008/top)

Cho hệ gồm N hành động. Mỗi hành động được biểu diễn như một bộ đôi <Si, Fi> tương ứng với thời gian bắt đầu và thời gian kết thúc của mỗi hành động. Hãy tìm phương án thực hiện nhiều nhất các hành động được thực hiện bởi một máy hoặc một người sao cho hệ không xảy ra mâu thuẫn.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số lượng hành động N; dòng tiếp theo đưa vào N số Si tương ứng với thời gian bắt đầu mỗi hành động; dòng cuối cùng đưa vào N số Fi tương ứng với thời gian kết thúc mỗi hành động; các số được viết cách nhau một vài khoảng trống.
* T, N, Si, Fi thỏa mãn ràng buộc: 1≤T≤100;  1≤N, Fi, Si≤1000.

**Output:**

* Đưa số lượng lớn nhất các hành động có thể được thực thi bởi một máy hoặc một người.

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

struct input{

int first,second;

};

input a[1010];

int n;

//sort theo thu tu ket thuc kthuc som hon dua len dau

bool cmp(input a,input b){

return a.second<b.second;

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i].first;

for(int i=0;i<n;i++) cin>>a[i].second;

sort(a,a+n,cmp);

int dem=1,i=0;

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

//neu cong viec dung sau co thoi gian bat dau >= tg ket thuc cua cong viec trc

if(a[j].first>=a[i].second){

dem++;

i=j;

}

}

cout<<dem<<endl;

}

return 0;

}

16. SẮP XẾP CÔNG VIỆC 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03009/top)

Cho N công việc. Mỗi công việc được biểu diễn như một bộ 3 số nguyên dương <JobId, Deadline, Profit>, trong đó JobId là mã của việc, Deadline là thời gian kết thúc của việc, Profit là lợi nhuận đem lại nếu hoàn thành việc đó đúng thời gian. Thời gian để hoàn toàn mỗi công việc là **1 đơn vị thời gian**. Hãy cho biết lợi nhuận lớn nhất có thể thực hiện các việc với giả thiết mỗi việc được thực hiện đơn lẻ.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là số lượng Job N; phần thứ hai đưa vào 3×N số tương ứng với N job.
* T, N, JobId, Deadline, Profit thỏa mãn ràng buộc:1≤T≤100;  1≤N≤1000; 1≤ JobId ≤1000; 1≤ Deadline ≤1000; 1≤ Profit ≤1000.

**Output:**

* Đưa số lượng công việc tương ứng và lợi nhuận lớn nhất có thể đạt được.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4 1 4 20  2 1 10  3 1 40  4 1 30 5 1 2 100  2 1 19  3 2 27  4 1 25  5 1 15 | 2 60 2 127 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

struct job{

int jobld,dealine,profit;

};

bool cmp(job a,job b){

return a.profit>b.profit;

}

int main(){

int t;cin>>t;

while(t--){

int n;cin>>n;

job a[n];

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

cin>>a[i].jobld>>a[i].dealine>>a[i].profit;

}

sort(a,a+n,cmp);

int ans=0,res=0;

int F[1005]={0};

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

while(F[a[i].dealine]&&a[i].dealine>0) a[i].dealine-=1;

if(!F[a[i].dealine]&&a[i].dealine>0){

F[a[i].dealine]=1;

res+=a[i].profit;

ans++;

}

}

cout<<ans<<" "<<res;

cout<<endl;

}

}

17. NỐI DÂY 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03011/top)

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j].

**Dữ liệu vào**

Dòng đầu ghi số bộ test T (T<10). Mỗi bộ test gồm 2 dòng. Dòng đầu tiên là số nguyên N (N ≤ 2\*106).

Dòng tiếp theo gồm N số nguyên dương c[i] ( 1 ≤ A[i] ≤ 109).

**Kết quả**

In ra đáp án của bộ test trên từng dòng, theo modulo 109+7.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 7  2 4 1 2 10 2 3 | 59 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

long long MOD=1e9+7,x;

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

//xay dung hang doi uu tien voi cac gia tri sx tang dan

priority\_queue<long long ,vector<long long>,greater<int>> q;

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

cin>>x;

q.push(x);

}

long long sum=0;

while(q.size()>1){

long long first=q.top();q.pop();

long long second=q.top();q.pop();

long long tmp=(first+second)%MOD;

q.push(tmp);

sum+=tmp;

sum%=MOD;

}

cout<<sum<<endl;

}

return 0;

}

18. SỐ NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03016/top)

Cho hai số nguyên dương S và D, trong đó S là tổng các chữ số và D là số các chữ số của một số. Nhiệm vụ của bạn là tìm số nhỏ nhất thỏa mãn S và D? Ví dụ với S = 9, D = 2 ta có số nhỏ nhất thỏa mãn S và D là 18.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ 2 số S và D được viết trên một dòng.
* T, S, D thỏa mãn ràng buộc: 1≤T≤100;  1≤ S,D≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Nếu không có đáp án, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 9 2 20 3 | 18 299 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int S,D;

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

string s;

cin>>S>>D;

if(S>9\*D) cout<<"-1"<<endl;

else{

S=S-1;

//khoi tao xau dang 10000

s+="1";

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

s+='0';

}

// ta se cong don tu cuoi

// neu co the la 9 thi s[i]=9

// neu <9 thi s[i]=so do roi break

// neu i=0 maf S>0 thi lay so du do cong them vowi 1 ban dau

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

if(i==0&&S>0) s[i]=(S+1)+'0';

else if(S>=9) s[i]='9';

else{

s[i]=S+'0';

break;

}

S-=(s[i]-'0');

}

cout<<s<<endl;

}

}

return 0;

}

19.GIÁ TRỊ NHỎ NHẤT CỦA XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03017/top)

Cho xâu ký tự S. Ta gọi giá trị của xâu S là tổng bình phương số lần xuất hiện mỗi ký tự trong S. Hãy tìm giá trị nhỏ nhất của xâu S sau khi thực hiện K lần loại bỏ ký tự.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là số K; phần thứ hai là một xâu ký tự S được viết trên một dòng.
* T, S, K thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000; 1≤K≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2  ABCCBC 2  AAAB | 6 2 |

#include<bits/stdc++.h>

using namespace std;

//vd ABCCBC A=1,B=2,C=3<k=2;

//c=2,k--,c or b=1 k--;

//a\*1+b\*2+c\*1=6

int k;

string s;

int main(){

int t;

cin >> t;

while(t--){

cin>>k>>s;

int dem[256]={0};

//dem so lan xuat hien cac ky tu trong chuoi

for(int i=0;i<s.size();i++) dem[s[i]]++;

//xay dung ham doi xap xep giam dan so lan xuat hien

priority\_queue<int,vector<int>> q;

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

if(dem[s[i]]>0){//neu ky tu xuat hien trong xau -> push so lan xh vao q( chi push 1 lan duy nhat)

q.push(dem[s[i]]);dem[s[i]]=0;

}

}

// dua ra so lan xuat hien nhieu nhat

// giam no di 1,push lai no vao q,dong thoi giam k di 1;

while(k>0&&q.size()>0){

k--;

int x=q.top();q.pop();

x--;

q.push(x);

}

long long sum=0;

while(q.size()>0){

int x=q.top();q.pop();

sum+=x\*x;

}

cout<<sum<<endl;

}

return 0;

}

20. SỐ MAY MẮN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03018/top)

Hoàng yêu thích các số may mắn. Ta biết rằng một số là số may mắn nếu biểu diễn thập phân của nó chỉ chứa các chữ số may mắn là 4 và 7. Ví dụ, các số 47, 744, 4 là số may mắn và 5, 17, 467 không phải. Hoàng muốn tìm số may mắn bé nhất có tổng các chữ số bằng n. Hãy giúp anh ấy

**Dữ liệu vào:**Dòng đầu ghi số bộ test, mỗi bộ test có một dòng chứa số nguyên n (1 ≤ n ≤ 106) — tổng các chữ số của số may mắn cần tìm.

**Kết quả:**In ra trên 1 dòng số may mắn bé nhất, mà tổng các chữ số bằng n. Nếu không tồn tại số thỏa mãn, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  11  10 | 47  -1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>n;

int dem4=0,dem7=0;

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

//duyet tu cuoi ve dem so luong so 4,7 cua xau tm

// xau cang nhieu so 4 thi gia gtri cang nho nen ta duyet tu n-1

if(n>=4\*i&&(n-4\*i)%7==0){

dem4=i;

dem7=(n-4\*i)/7;

}

}

//neu tim dc xau tm

if(4\*dem4+7\*dem7==n){

for(int i=0;i<dem4;i++) cout<<4;

for(int i=0;i<dem7;i++) cout<<7;

}

else cout<<"-1";

cout<<endl;

}

return 0;

}

21.  PHÂN SỐ ĐƠN VỊ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03019/top)

Một phân số đơn vị nếu tử số của phân số đó là 1. Mọi phân số nguyên dương đều có thể biểu diễn thành tổng các phân số đơn vị. Ví dụ 2/3 = 1/2  + 1/6. Cho phân số nguyên dương P/Q bất kỳ (P < Q), hãy biểu diễn phân số nguyên dương thành tổng phân số đơn vị với số hạng tử là ít nhất.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ đôi tử số P và mẫu số Q của phân số nguyên dương được viết trên một dòng.
* T, P, Q thỏa mãn ràng buộc: 1≤T≤100;  1≤P, Q≤100.

**Output:**

* Đưa ra đáp án tìm được trên 1 dòng, theo dạng “1/a + 1/b + …”

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  2 3  1 3 | 1/2 + 1/6  1/3 |

#include<bits/stdc++.h>

using namespace std;

//gt code:

//2/3 va 1/3->1/(3/2)va 1/(3/1)<1/(mau/tu);

//neu mau%tu==0 dung lai va in kq(1/3)

//neu ko gia su 1/(3/2)=1/a=1/b+x; <b:res,a:mau/tu>

//ta thay 1/a>1/b->b>a ->gan luon ban dau=a+1 lay nguyen

//luc do x=1/a-1/b=b-a/a\*b, x dung lai khi a\*b%b-a;

//vd 1/(3/2) |3/2+1|=2 ->b=2->x=2/3-1/2=1/6 dung

int main(){

int t;

cin >> t;

while(t--){

long long tuso,mauso;

cin>>tuso>>mauso;

long long res;

while(1){

if(mauso%tuso==0){

cout<<"1/"<<mauso/tuso<<endl;

break;

}

else{

res=mauso/tuso+1;

cout<<"1/"<<res;

cout<<" + ";

tuso=tuso\*res-mauso;

mauso=mauso\*res;

}

}

}

return 0;

}

22. BIỂU THỨC ĐÚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA03020/top)

Cho một mảng S gồm 2×N ký tự, trong đó có N ký tự ‘[’ và N ký tự ‘]’. Xâu S được gọi là viết đúng nếu S có dạng S2[S1] trong đó S, S2 là các xâu viết đúng. Nhiệm vụ của bạn là tìm số các phép đổi chỗ ít nhất các ký tự kề nhau của xâu S viết sai để  S trở thành viết đúng. Ví dụ với xâu S =”[]][][” ta có số phép đổi chỗ kề nhau ít nhất là 2.

**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 xâu S viết sai theo nguyên tắc kể trên.
* T, S thòa mãn ràng buộc: 1≤T≤100; 1≤length(S)≤100000.

**Output:**

* Đưa kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  []][][  [][][] | 2 0 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

stack<char> ans;

int tm=0,res=0;

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

if(ans.size()==0) ans.push(s[i]);

else if(ans.top()=='['&&s[i]==']'){

ans.pop();

tm+=2;

}

else if(ans.top()==']'&&s[i]=='['){

res+=ans.size()+tm;

ans.pop();

}

else if(s[i]=='[') ans.push(s[i]);

else if(ans.top()==']'&&s[i]==']') ans.push(s[i]);

if(ans.size()==0) tm=0;

}

cout<<res<<endl;

}

}

Dạng 4 : CHIA ĐỂ TRỊ

1. LŨY THỪA

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04001/top)

Cho số nguyên dương N và K. Hãy tính NK modulo 109+7.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 1 số nguyên N và K (1 ≤ N ≤ 1000, 1 ≤ K ≤ 109).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 3  4 2 | 8  16 |

#include<bits/stdc++.h>

using namespace std;

long long MOD=1e9+7;//chia de tri

long long cd(long long n,long long k){

long long res=0;

if(k==1) res=n;

else{

res=cd(n,k/2);

res=(res\*res)%MOD;

if(k%2!=0) res=(res\*n)%MOD;

}

return res;

}

int main(){

int t;

cin >> t;

while(t--){

long long n,k;cin>>n>>k;

cout<<cd(n,k)<<endl;

}

return 0;

}

2.  GẤP ĐÔI DÃY SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04004/top)

Một dãy số tự nhiên bắt đầu bởi con số 1 và được thực hiện N-1 phép biến đổi “gấp đôi” dãy số như sau:

Với dãy số A hiện tại, dãy số mới có dạng A, x, A trong đó x là số tự nhiên bé nhất chưa xuất hiện trong A.

Ví dụ với 2 bước biến đổi, ta có [1] à [1 2 1] à [1 2 1 3 1 2 1].

Các bạn hãy xác định số thứ K trong dãy số cuối cùng là bao nhiêu?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm số nguyên dương N và K (1 ≤ N ≤ 50, 1 ≤ K ≤ 2N - 1).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3 2  4 8 | 2  4 |

//by hung nguyen

#include<bits/stdc++.h>

//gt code vd 4 9

//1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

//do dai xau=2^n;tai vi tri trung tam 2^n/2

//tai vi tri 9 coi vi tri 8(4) la trung tam thi bang vi tri 7

//tai vi tri 7 xet nua xau dang trc(do dai 8) coi vi tri 4(3) la trung tam thi bang vi tri 1

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;long long k;cin>>n>>k;

long long mid=pow(2,n)/2;//vi tri phan tu chinh giua cua chuoi

while(1){

if(n==1){//xau co do dai 1

cout<<1;break;

}

if(k==mid){//k tai phan tu chinh giua

cout<<n;break;

}

if(k>mid){//k>mid thi lay doi xung de dua ve vi tri so do ban dau

k=mid-(k-mid);

}

n--;

mid/=2;

}

cout<<endl;

}

return 0;

}

3.  ĐẾM DÃY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04003/top)

Cho số nguyên dương n. Hãy cho biết có bao nhiêu dãy số nguyên dương có tổng các phần tử trong dãy bằng n.

**Dữ liệu vào:**dòng đầu tiên chứa số nguyên T là số bộ dữ liệu, mỗi bộ dữ liệu ghi một số nguyên dương n duy nhất không qua 1018.

**Kết quả:**Mỗi bộ dữ liệu ghi ra một số nguyên duy nhất là số dư của kết quả tìm được khi chia cho **123456789**.

**Ví dụ:**

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

// by hung nguyen

#include <bits/stdc++.h>

const long long mod = 123456789;

using namespace std;

long long n, k;

long long solution(long long n){

if (n == 1)

return 2;

long long x = solution(n / 2);

if(n % 2 == 0)

return (x % mod \* x % mod) % mod;

else

return (2 \* x % mod \* x % mod) % mod;

}

int main(){

int t;

cin >> t;

while(t--){

cin >> n;

cout << solution(n-1) << endl;

}

return 0;

}

4. DÃY XÂU FIBONACI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04005/top)

Một dãy xâu ký tự G chỉ bao gồm các chữ cái A và B được gọi là dãy xâu Fibonacci nếu thỏa mãn tính chất:    G(1) = A; G(2) = B; G(n) = G(n-2)+G(n-1). Với phép cộng (+) là phép nối hai xâu với nhau.  Bài toán đặt ra là tìm ký tự ở vị trí thứ i (tính từ 1) của xâu Fibonacci thứ n.

**Dữ liệu vào:**Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng 2 số nguyên N và i (1<N<93). Số i đảm bảo trong phạm vi của xâu G(N) và không quá 18 chữ số. **Kết quả:** Ghi ra màn hình kết quả tương ứng với từng bộ test.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  6 4  8 19 | A  B |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long F[100],n,k;

void xuly(){

while(n>2){

if(k<=F[n-2]) n-=2;

else{

k-=F[n-2];

n-=1;

}

}

if(n==1) cout<<'A';

else cout<<'B';

}

int main(){

int t;

cin >> t;

F[1]=F[2]=1;

for(int i=3;i<=93;i++) F[i]=F[i-1]+F[i-2];

while(t--){

cin>>n>>k;

xuly();

cout<<endl;

}

return 0;

}

5. LŨY THỪA ĐẢO

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04002/top)

Cho mảng số N. Ta gọi số đảo của N là R. Hãy tìm lũy thừa R của N. Đưa ra kết quả của bài toán dưới dạng modulo với 109 + 7.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm là số N được ghi trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤1010.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2 12 | 4 864354781 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long mod=1e9+7;//chia de tri

long long cd(long long n,long long k){

if(k== 1)

return n;

long long tmp= cd(n, k/2);

if (k%2==0)

return (tmp % mod \* tmp % mod) % mod;

else

return n \* (tmp % mod \* tmp % mod) % mod;

}

int main(){

int t;

cin >> t;

while(t--){

long long n;cin>>n;

long long x=n,k=0;

while(x>0){

k=k\*10+x%10;

x/=10;

}

cout<<cd(n,k)<<endl;

}

return 0;

}

6. DÃY CON LIÊN TIẾP CÓ TỔNG LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04010/top)

Cho mảng A[] gồm N số có cả các số âm và số dương. Nhiệm vụ của bạn là tìm mảng con liên tục có tổng lớn nhất của mảng. Ví dụ với mảng A[]={-2, -5, 6, -2,-3, 1, 5, -6} ta có kết quả là 7 tương ứng với dãy con {6, -2, -3, 1, 5}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào hai số N tương ứng với số phần tử của mảng; dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N≤100; -100≤A[i] ≤100.

**Output:**

* Đưa ra tổng con liên tục lớn nhất của mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 1 8 -2 -5 6 -2 -3 1 5 -6 | 7 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n],ok=0,max=a[0];

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

cin>>a[i];

if(a[i]>0) ok=1;

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

}

if(ok==0) cout<<max<<endl;//mang toan so am -> lay so am lon nhat

else{

int sum=0,tmp=a[0];

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

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

sum=0;

}

else sum+=a[i];

if(tmp<sum) tmp=sum;

}

cout<<tmp<<endl;

}

}

return 0;

}

7. TÍNH FLOOR(X)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04015/top)

Cho mảng đã được sắp xếp A[] gồm N phần tử không có hai phần tử giống nhau và số X. Nhiệm vụ của bạn là tìm floor(X). Trong đó, K=floor(X) là phần tử lớn nhất trong mảng A[] nhỏ hơn hoặc bằng X.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số N là số phần tử của mảng A[] và số X; dòng tiếp theo đưa vào N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N≤107; 1≤ A[i]≤1018.

**Output:**

* Đưa ra vị trí của  floor(X) trong mảng A[] hoặc -1 nếu không tồn tại floor(X) của mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 7 0 1 2 8 10 11 12 19 7 5 1 2 8 10 11 12 19 7 10 1 2 8 10 11 12 19 | -1 2 4 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

long long a[n],X,dem=-1,tmp=0;

cin>>X;

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

cin>>a[i];

if(a[i]<=X&&a[i]>tmp){

tmp=a[i];

dem=i+1;

}

}

cout<<dem<<endl;

}

return 0;

}

8.  PHẦN TỬ THỨ K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04016/top)

Cho hai mảng đã được sắp xếp A[], B[] gồm M, N phần tử theo thứ tự và số K. Nhiệm vụ của bạn là tìm phần tử ở vị trí số K sau khi trộn hai mảng để nhận được một mảng được sắp xếp.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số M, N, K; dòng tiếp theo đưa vào M số của mảng A[];dòng tiếp theo đưa vào N số của mảng B[];các số được viết cách nhau một vài khoảng trống.
* T, M,N, A[i], B[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N, A[i], B[i]≤106; 1≤ K≤N+M.

**Output:**

* Đưa ra giá trị phần tử thứ K của mỗi test theo từng dòng.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,m,k;cin>>n>>m>>k;

int a[n+m];

for(int i=0;i<n+m;i++) cin>>a[i];

sort(a,a+n+m);

cout<<a[k-1]<<endl;

}

return 0;

}

9. PHẦN TỬ KHÁC NHAU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04017/top)

Cho hai mảng đã được sắp xếp A[] và B[] gồm N và N-1 phần tử. Các phần tử của mảng A[] chỉ khác mảng B[] một phần tử duy nhất. Hãy tìm vị trí của phần tử khác nhau giữa A[] và B[].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số N; dòng tiếp theo đưa vào N số của mảng A[];dòng tiếp theo đưa vào N-1 số của mảng B[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i], B[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N≤107; 0≤ A[i]≤1018.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

long long a[n],b[n],ok=1,vt;

for(int i=0;i<n;i++)cin>>a[i];

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

cin>>b[i];

if(ok==1&&b[i]!=a[i]){

ok=0;vt=i+1;

}

}

cout<<vt<<endl;

}

return 0;

}

10. ĐẾM SỐ 0

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04018/top)

Cho mảng A[] gồm N phần tử chỉ bao gồm các số 0 và 1. Các số 0 được đặt trước các số 1. Hãy đếm các số 0 với thời gian log(N).

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số N; dòng tiếp theo đưa vào N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N≤1000; 0≤ A[i]≤1.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 3 12 0 0 0 0 0 0 0 0 0 1 1 1  5 0 0 0 0 0 6 1 1 1 1 1 1 | 9 5 0 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

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

if(a[n-1]==0) cout<<n;

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

cout<<i;break;

}

}

cout<<endl;

}

return 0;

}

11.  TÌM KIẾM NHỊ PHÂN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04020/top)

**BÀI 22. TÌM KIẾM NHỊ PHÂN**

Cho dãy số A[] gồm có N phần tử đã được sắp xếp tăng dần và số K.

Nhiệm vụ của bạn là kiểm tra xem số K có xuất hiện trong dãy số hay không. Nếu có hãy in ra vị trí trong dãy A[], nếu không in ra “NO”.

**Input:**

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

Mỗi test bắt đầu bằng số nguyên N và K (N ≤ 100 000, 0 ≤ K ≤ 106).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 106), các phần tử là riêng biệt.

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  5 3  1 2 3 4 5  6 5  0 1 2 3 9 10 | 3  NO |

//dfiodfjfjeop

#include<bits/stdc++.h>

using namespace std;

long long slove(long long a[],long long k,long long l,long long r){

if(r>=l){

long long s = l+(r-l)/2;

if(a[s] == k) return s;

if(a[s] > k ) return slove(a,k,l,s-1);

if(a[s]<k) return slove(a,k,s+1,r);

}

return -1;

}

main(){

int t;cin>>t;

while(t--){

long long n,k;cin>>n>>k;

long long a[n];long long l=0;

for(long long i=0;i<n;i++) cin>>a[i];

long long kq = slove(a,k,l,n-1);

if(kq==-1) cout<<"NO";

else cout<<kq + 1;

cout<<endl;

}

}

12.0. DÃY XÂU NHỊ PHÂN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04037/top)

Người ta tạo ra một dãy xâu ký tự nhị phân X[] trong đó:

X[1] = “0”

X[2] = “1”

X[n] = X[n-2] + X[n-1] với n>2

Với phép cộng (+) là phép nối hai xâu với nhau.

Cho hai số tự nhiên N và K (1<N<93; K đảm bảo trong phạm vi của xâu X[N]).

Hãy xác định ký tự thứ K trong xâu X[N] là ký tự ‘0’ hay ký tự ‘1’.

**Input:**Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng 2 số nguyên N và K.

**Output:** Ghi ra màn hình kết quả tương ứng với từng bộ test.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  6 4  8 19 | 0  1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long F[100],n,k;

void xuly(){

while(n>2){

if(k<=F[n-2]) n-=2;

else{

k-=F[n-2];

n-=1;

}

}

if(n==1) cout<<'0';

else cout<<'1';

}

int main(){

int t;

cin >> t;

F[1]=F[2]=1;

for(int i=3;i<=93;i++) F[i]=F[i-1]+F[i-2];

while(t--){

cin>>n>>k;

xuly();

cout<<endl;

}

return 0;

}

12.  HỆ CƠ SỐ K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04007/top)

Cho hai số A, B ở hệ cơ số K. Hãy tính tổng hai số đó ở hệ cơ số K.

**Input:** Dòng đầu ghi số bộ test T (T<10). Mỗi bộ test ghi 3 số K,A,B.

            (2≤K≤10; A và B nếu biểu diễn trong hệ cơ số 10 đều nhỏ hơn 109)

**Output:** In ra tổng của A và B trong hệ cơ số K

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2 1 10 | 11 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int k;

string a,b,res;

int main(){

int t;

cin >> t;

while(t--){

cin>>k>>a>>b;

while(a.size()<b.size()) a="0"+a;

while(a.size()>b.size()) b="0"+b;

int nho=0;

res="";

// cong tu cuoi -> dau va duy tri bien nho

for(int i=a.size()-1;i>=0;i--){

int x=int(a[i]-'0')+int (b[i]-'0')+nho;

res=char(x%k+'0')+res;//cong dan ve dau chuoi

nho=x/k;

}

if(nho>0) res=char(nho+'0')+res;

cout<<res<<endl;

}

return 0;

}

13. SẮP XẾP KANGURU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04013/top)

Có N con kanguru trong vườn thú, con thứ i có chiều cao bằng A[i]. Con kanguru có chiều cao X có thể chứa được một con có chiều cao bằng Y trong túi của nó nếu như X >= 2\*Y.

Một con đã chứa một con kanguru rồi, thì không thể nhảy vào túi một con kanguru khác.

Bầy Kanguru rất thích chơi trốn tìm, vì vậy chúng thường xuyên nhảy vào túi của nhau. Các bạn hãy tính toán xem trong trường hợp tối ưu, số con kanguru nhìn thấy trong vườn thú ít nhất bằng bao nhiêu?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

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

Dòng tiếp theo gồm N số nguyên A[i] (1 ≤ A[i] ≤ 100 000).

**Output:**

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

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n],dem=n;

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

for(int i=n-1,j=n/2-1;i>=n/2&&j>=0;){

if(a[i]>=2\*a[j]){

dem--;i--;j--;

}

else j--;

}

cout<<dem<<endl;

}

return 0;

}

14. SỐ FIBONACCI THỨ N

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04008/top)

Dãy số Fibonacci được xác định bằng công thức như sau:

F[0] = 0, F[1] = 1;

F[n] = F[n-1] + F[n-2] với mọi n >= 2.

Các phần tử đầu tiên của dãy số là 0, 1, 1, 2, 3, 5, 8, ...

Nhiệm vụ của bạn là hãy xác định số Fibonaci thứ n. Do đáp số có thể rất lớn, in ra kết quả theo modulo 109+7.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 1000).

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

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  2  6  20 | 1  8  6765 |

 // by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long mod=1e9+7,n,F[2][2],M[2][2];

void mu(long long f[2][2],long long m[2][2]){

long long x=(f[0][0]\*m[0][0]%mod+f[0][1]\*m[1][0]%mod)%mod;

long long y=(f[0][0]\*m[0][1]%mod+f[0][1]\*m[1][1]%mod)%mod;

long long z=(f[1][0]\*m[0][0]%mod+f[1][1]\*m[1][0]%mod)%mod;

long long t=(f[1][0]\*m[0][1]%mod+f[1][1]\*m[1][1]%mod)%mod;

F[0][0]=x;F[0][1]=y;F[1][0]=z;F[1][1]=t;

}

void pow(long long f[2][2],long long n){

if(n<=1) return;

pow(f,n/2);mu(f,f);

if(n&1) mu(f,M);

}

void init(){

F[0][0]=F[0][1]=F[1][0]=1;

F[1][1]=0;

M[0][0]=M[0][1]=M[1][0]=1;

M[1][1]=0;

if(n==0) cout<<0;

else{

pow(F,n-1);

cout<<F[0][0];

}

cout<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

init();

}

return 0;

}

15. LŨY THỪA MA TRẬN 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04009/top)

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 8 5  5 3  597240088 35500972 473761863  781257150 154135232 527013321  965274212 272769492 580264779 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

long long k,a[10][10],b[10][10],MOD=1e9+7;

//ham nhan ma tran

void Xuly(long long x[10][10],long long y[10][10]){

long long tmp[10][10];

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

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

long long ans=0;

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

ans+=(x[i][k]\*y[k][j])%MOD;

ans%=MOD;

}

tmp[i][j]=ans;

}

}

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

for(int j=0;j<n;j++) a[i][j]=tmp[i][j];

}

//su dung chia de tri

void nhanMatrix(long long a[10][10],int n){

if(n<=1) return ;

nhanMatrix(a,n/2);

Xuly(a,a);

if(n%2==1) Xuly(a,b);//neu n le thi nhan them ma tran b(=a)

}

//in

void in(){

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

for(int j=0;j<n;j++) cout<<a[i][j]<<" ";

cout<<endl;

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>k;

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

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

cin>>a[i][j];

b[i][j]=a[i][j];

}

nhanMatrix(a,k);

in();

}

return 0;

}

16. TÍCH ĐA THỨC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04012/top)

Cho hai đa thức P và Q được biểu diễn như một mảng bao gồm các hệ số của đa thức. Ví dụ với P(x) = 5 + 0x1 +10x2 + 6x3 được biểu diễn như mảng P[] ={5, 0, 10, 6}. Hãy đưa ra đa thức R = P×Q theo các hệ số của R với cách biểu diễn như trên.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào hai số M, N tương ứng với lũy thừa lớn nhất của đa thức P và Q; dòng tiếp theo đưa vào M số là hệ số của đa thức P; dòng cuối cùng đưa vào M số là hệ số của đa thức Q.
* T, M, N, P[i], Q[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤M, N≤100; 1≤P[i], Q[i]≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4 3 1 0 3 2 2 0 4 5 4 1 9 3 4 7 4 0 2 5 | 2 0 10 4 12 8 4 36 14 39 79 23 34 35 |

#include<bits/stdc++.h>

using namespace std;

//mu 0 1 2 3

//p 1 0 3 2

//q 2 0 4

//mu 0 1 2 3 4 5

//hs 1\*2/1\*0+0\*2/0\*0+3\*2+4\*2/3\*0+4\*1+2\*2/3\*4+2\*0/2\*4

int n,m;

struct data{

int heso,mu;

};

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>m;

vector<vector<data>> res;

data a[n],b[m];

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

cin>>a[i].heso;

a[i].mu=i;

}

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

cin>>b[j].heso;

b[j].mu=j;

}

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

vector<data> x;

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

x.push\_back({a[i].heso\*b[j].heso,a[i].mu+b[j].mu});

}

res.push\_back(x);

}

int mu=0;

while(mu<m+n-1){

int tmp=0;

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

for(int j=0;j<res[i].size();j++){

if(res[i][j].mu==mu) tmp+=res[i][j].heso;

}

}

cout<<tmp<<" ";

mu++;

}

cout<<endl;

}

return 0;

}

17. TÍCH HAI SỐ NHỊ PHÂN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04011/top)

Cho hai xâu nhị phân biểu diễn hai số. Nhiệm vụ của bạn là đưa ra tích của hai số. Ví dụ với xâu S1=”1100” và S2=”1010” ta sẽ có kết quả là 120.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 hai xâu nhị phân S1, S2 được viết trên một dòng.
* T, S1, S2 thỏa mãn ràng buộc: 1≤T≤100;  1≤ length(S1), length(S2)≤30.

**Output:**

* Đưa ra tích của mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 1100 01 01 01 | 12  1 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

string s1,s2;

long long p,sum1,sum2;

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>s1>>s2;

p=1,sum1=0;sum2=0;

// chuyen doi hai chuoi ve dang thap phan

for(int i=s1.size()-1;i>=0;i--){

sum1+=(int)(s1[i]-'0')\*p;

p\*=2;

}

// khoi tao lai p

p=1;

for(int i=s2.size()-1;i>=0;i--){

sum2+=(int)(s2[i]-'0')\*p;

p\*=2;

}

// nhan hai so o dang thap phan

cout<<sum1\*sum2<<endl;

}

return 0;

}

18.  DÃY XÂU NHỊ PHÂN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04026/top)

Người ta tạo ra một dãy xâu ký tự nhị phân X[] trong đó:

X[1] = “0”

X[2] = “1”

X[n] = X[n-2] + X[n-1] với n>2

Với phép cộng (+) là phép nối hai xâu với nhau.

Cho hai số tự nhiên N và K (1<N<93; K đảm bảo trong phạm vi của xâu X[N]).

Hãy xác định ký tự thứ K trong xâu X[N] là ký tự ‘0’ hay ký tự ‘1’.

**Input:**Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng 2 số nguyên N và K.

**Output:** Ghi ra màn hình kết quả tương ứng với từng bộ test.

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long F[100],n,k;

void xuly(){

while(n>2){

if(k<=F[n-2]) n-=2;

else{

k-=F[n-2];

n-=1;

}

}

if(n==1) cout<<'0';

else cout<<'1';

}

int main(){

int t;

cin >> t;

F[1]=F[2]=1;

for(int i=3;i<=93;i++) F[i]=F[i-1]+F[i-2];

while(t--){

cin>>n>>k;

xuly();

cout<<endl;

}

return 0;

}

19. ĐẾM SỐ BÍT 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04006/top)

Cho số nguyên dương N. Mỗi bước, bạn sẽ biến đổi N thành [N/2], N mod 2, [N/2]. Sau khi thực hiện một cách triệt để, ta thu được một dãy số chỉ toàn số 0 và 1.

Nhiệm vụ của bạn là hãy đếm các số bằng 1 trong đoạn [L, R] của dãy số cuối cùng.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 3 số nguyên N, L, R (1 ≤ N, L, R < 250, 0 ≤ R-L ≤ 100 000).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  7 2 5  10 3 10 | 4  5 |

20. CẶP NGHỊCH THẾ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04014/top)

Cho mảng A[] gồm N phần. Ta gọi cặp nghịch thế của mảng A[] là số các cặp i, j sao cho i<j và A[i]>A[j]. Đối với mảng đã được sắp xếp thì số cặp nghịch thế bằng 0. Mảng đã sắp theo thứ tự giảm dần có số đảo ngược cực đại. Nhiệm vụ của bạn là hãy đưa ra số cặp nghịch thế của mảng A[] gồm N phần tử.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N tương ứng với số phần tử của mảng A[]; phần thứ 2 là N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤107; 1≤A[i]≤1018.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 5 2 4 1 3 5  5  5 4 3 2 1 | 3  10 |

21. LŨY THỪA MA TRẬN 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04021/top)

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. **Sau đó, tính tổng các phần tử trên đường chéo chính.** Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 11  331640092 |

22. LŨY THỪA MA TRẬN 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04022/top)

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. Sau đó, **tính tổng các phần tử trên đường chéo phụ.** Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 10  593171300 |

23. LŨY THỪA MA TRẬN 4

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. Sau đó, **tính tổng các phần tử của hàng đầu tiên**. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 13  106502916 |

24. LŨY THỪA MA TRẬN 5

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04024/top)

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. Sau đó, **tính tổng các phần tử của hàng cuối cùng**. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 8  818308476 |

25. LŨY THỪA MA TRẬN 6

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04025/top)

Cho ma trận vuông A kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận X = AK với K là số nguyên cho trước. Sau đó, **tính tổng các phần tử của cột đầu tiên**. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 109+7.

**Input:**

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

Mỗi test bắt gồm một số nguyên N và K (1 ≤ N ≤ 10, 1 ≤ K ≤ 109) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  2 5  1 1  1 0  3 1000000000  1 2 3  4 5 6  7 8 9 | 13  343771436 |

26. CẶP ĐIỂM GẦN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA04019/top)

Cho N điểm trên mặt phẳng tọa độ Oxy. Bạn cần tìm khoảng cách ngắn nhất giữa hai điểm trong số N điểm đã cho.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi một số nguyên N (1 ≤ N ≤ 100 000).

N dòng tiếp theo, mỗi dòng gồm 2 số nguyên X[i], Y[i] (-106 ≤ X[i], Y[i] ≤ 106).

**Output:**

Với mỗi test, in ra đáp án trên một dòng với **độ chính xác 6 chữ số sau dấu phẩy.**

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  6  2 3  12 30  40 50  5 1  12 10  3 4  3  0 0  3 0  4 0 | 1.414214  1.000000 |

Dạng 5: Quy Hoạch Động

1. XÂU CON CHUNG DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05001/top)

Cho 2 xâu S1 và S2. Hãy tìm xâu con chung dài nhất của 2 xâu này (các phần tử không nhất thiết phải liên tiếp nhau).

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 20). Mỗi test gồm hai dòng, mô tả xâu S1 và S2, mỗi xâu có độ dài không quá 1000 và chỉ gồm các chữ cái in hoa.

**Output:**Với mỗi test, in ra độ dài dãy con chung dài nhất trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  AGGTAB  GXTXAYB  AA  BB | 4  0 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

string s1,s2;

int F[2000][2000];

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>s1>>s2;

int dem=0;

memset(F,0,sizeof(F));

for(int i=1;i<=s1.size();i++){

for(int j=1;j<=s2.size();j++){

//luc dau F[i][j]=0 khi gap so lap dau tien, tang len 1

// khi gap cac th lap tiep theo cong voi so trc

if(s1[i-1]==s2[j-1]) F[i][j]=F[i-1][j-1]+1;

//neu hai so tai vi tri i,j ko trung nhau

//ta lay gtri max tai vi tri trcx do

//vd trung dau tien tai vtri (2 3)

// neu cac vi tri tiep theo 2 4 va 3 3ko trung thi duy tri =1

//vd tai 4 4 lai xay ra trung thi ta lay f(3 3)+1=2

// duy tri tai cac vi tri ko trung tiep theo =2

else F[i][j]=max(F[i][j-1],F[i-1][j]);

//cap nhat dem

dem=(dem,F[i][j]);

}

}

cout<<dem<<endl;

}

return 0;

}

2. DÃY CON TĂNG DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05004/top)

Cho một dãy số nguyên gồm N phần tử A[1], A[2], ... A[N].

Biết rằng dãy con tăng là 1 dãy A[i1],... A[ik]

thỏa mãn i1 < i2 < ... < ik và A[i1] < A[i2] < .. < A[ik].

Hãy cho biết dãy con tăng dài nhất của dãy này có bao nhiêu phần tử?

**Input:**Dòng 1 gồm 1 số nguyên là số N (1 ≤ N ≤ 1000). Dòng thứ 2 ghi N số nguyên A[1], A[2], .. A[N] (1 ≤ A[i] ≤ 1000).

**Output:** Ghi ra độ dài của dãy con tăng dài nhất.

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[1001],F[1001],res;

//vd 1 2 5 4 6 2

//i 1 2 3 4 5 6

//for i=1,F[1]=1

//for i=2,F[2]=1 / F[2]=max(F[2],F[1]+1)=2

//for i=3,F[3]=1/F[3]=max(F[3],F[1]+1)=2/F[3]=max(F[3],F[2]+1)=3

//tuong tu cac th khac

int main(){

cin>>n;

res=0;

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

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

F[i]=1;

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

if(a[i]>a[j]) F[i]=max(F[i],F[j]+1);

}

res=max(res,F[i]);

}

cout<<res<<endl;

return 0;

}

3. DÃY CON CÓ TỔNG BẰNG S

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05008/top)

Cho N  số nguyên dương tạo thành dãy A={A1, A2, ..., AN}. Tìm ra một dãy con của dãy A (không nhất thiết là các phần tử liên tiếp trong dãy) có tổng bằng S cho trước.

**Input:**Dòng đầu ghi số bộ test T (T<10).  Mỗi bộ test có hai dòng, dòng đầu tiên ghi hai số nguyên dương N và S (0 < N ≤ 200) và S (0 < S ≤ 40000). Dòng tiếp theo lần lượt ghi N số hạng của dãy A là các số A1, A2, ..., AN (0 < Ai≤ 200).

**Output:**  Với mỗi bộ test, nếu bài toán vô nghiệm thì in ra “NO”, ngược lại in ra “YES”

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 6 1 2 4 3 5  10 15  2 2 2 2 2 2 2 2 2 2 | YES  NO |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

//vd test 1 3 5 7 2 s=11

//for i=1 F[1]=1/ j chay 11->1 F[j-x]=0

//for i=2 F[3]=1/ j chay tu 11->7 F[j-x]=0/j=6 F[j-x]=F[6-3]=1->F[6]=1 tuong tu F[4-3]=1->F[4]=1;

//for i=3 F[7]=1/ tai j=11 F[11-7]=F[4]=1->F[11]=1->co the tao ra day co tong =11

int n,s;

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>s;

int F[40001]={0},x;

F[0]=1;

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

cin>>x;

//danh dau F tai i =1;

F[x]=1;

for(int j=s;j>=x;j--){

if(F[j-x]==1) F[j]=1;

}

}

if(F[s]==1) cout<<"YES";

else cout<<"NO";

cout<<endl;

}

return 0;

}

4. TỔ HỢP C(n, k)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05012/top)

Cho 2 số nguyên n, k. Bạn hãy tính C(n, k) modulo 109+7.

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test gồm 2 số nguyên n, k (1 ≤ k ≤ n ≤ 1000).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 2  10 3 | 10  120 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

long long P=1e9+7;

long long c[1001][1001];

int n,r;

void tohop(){

for(int i=0;i<=1000;i++){

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

if(i==0||j==0) c[i][j]=1;

else c[i][j]=(c[i-1][j]+c[i-1][j-1])%P;

}

}

}

int main(){

int t;

cin >> t;

tohop();

while(t--){

int n,r;

cin>>n>>r;

cout<<c[n][r]<<endl;

}

return 0;

}

5. BẬC THANG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05013/top)

Một chiếc cầu thang có N bậc. Mỗi bước, bạn được phép bước lên trên tối đa K bước. Hỏi có tất cả bao nhiêu cách bước để đi hết cầu thang? (Tổng số bước đúng bằng N).

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 100).
* Mỗi test gồm hai số nguyên dương N và K(1 ≤ N ≤ 100000, 1 ≤ K ≤ 100).

**Output:**

* Với mỗi test, in ra đáp án tìm được trên một dòng theo modulo 109+7.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2 2  4 2 | 2  5 |

#include<bits/stdc++.h>

using namespace std;

long long MOD=1e9+7;

int main(){

int t;

cin >> t;

while(t--){

int n,k;

cin>>n>>k;

long long F[n+1]={0};

F[0]=F[1]=1;

//F[n]=F[n-1]+F[n-2]+...+F[n-k];

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

for(int j=1;j<=min(i,k);j++){

F[i]+=F[i-j];

F[i]%=MOD;

}

}

cout<<F[n]<<endl;

}

return 0;

}

6. DÃY CON LẶP LẠI DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05002/top)

Cho xâu ký tự S. Nhiệm vụ của bạn là tìm độ dài dãy con lặp lại dài nhất trong S. Dãy con có thể chứa các phần tử không liên tiếp nhau.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào độ dài xâu str; dòng tiếp theo đưa vào xâu S.
* T, str thỏa mãn ràng buộc: 1 ≤ T ≤ 100;  1 ≤ size(S) ≤ 100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 3 abc  5 axxxy | 0 2 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

string s;cin>>s;

int F[n+1][n+1];

memset(F,0,sizeof(F));

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

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

if(s[i-1]==s[j-1]&&i!=j)

F[i][j]=F[i-1][j-1]+1;

else

F[i][j]=max(F[i-1][j],F[i][j-1]);

}

}

cout<<F[n][n]<<endl;

}

return 0;

}

7. SỐ BƯỚC ÍT NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05005/top)

Cho mảng A[] gồm N số nguyên. Nhiệm vụ của bạn là sắp xếp lại mảng số với số lượng bước là ít nhất. Tại mỗi bước, bạn chỉ được phép chèn phần tử bất kỳ của mảng vào vị trí bất kỳ trong mảng. Ví dụ A[] = {2, 3, 5, 1, 4, 7, 6 }sẽ cho ta số phép chèn ít nhất là 3 bằng cách lấy số 1 chèn trước số 2, lấy số 4 chèn trước số 5, lấy số 6 chèn trước số 7 ta nhận được mảng được sắp.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là một số N; dòng tiếp theo đưa vào N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤1000; 1≤A[i] ≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

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

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int main() {

int t; cin >> t;

while (t--) {

int n; cin>>n;

int a[1001];

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

cin>>a[i];

}

int lenMax = 0;

int b[1001]={0};

b[0]=1;

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

b[i]=1;

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

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

b[i]=max(b[i],b[j]+1);

}

}

lenMax=max(lenMax,b[i]);

}

cout<<n-lenMax<<endl;

}

return 0;

}

8. CON ẾCH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05025/top)

Một con ếch có thể nhảy 1, 2, 3 bước để có thể lên đến một đỉnh cần đến. Hãy đếm số các cách con ếch có thể nhảy đến đỉnh.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là số n là số bước con ếch có thể lên được đỉnh.
* T, n thỏa mãn ràng buộc: 1≤T≤100;  1≤n ≤50.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 1 5 | 1 13 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;

cin>>n;

long long F[n+1]={0};

F[0]=F[1]=1;

//F[n]=F[n-1]+F[n-2]+...+F[n-k];

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

for(int j=1;j<=min(i,3);j++){

F[i]+=F[i-j];

}

}

cout<<F[n]<<endl;

}

return 0;

}

9. XEM PHIM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05026/top)

John có một đàn bò. Một ngày đẹp trời, anh ta quyết định mua xe tải với khả năng chở được C kg (1000 ≤ C ≤ 25000) để đưa những con bò đi xem phim. Cho số con bò là N (20 ≤ N ≤ 100) và khối lượng w[i] của từng con (đều nhỏ hơn C), hãy cho biết **khối lượng bò lớn nhất** mà John có thể đưa đi xem phim là bao nhiêu.

**Input:**

* Dòng 1: 2 số nguyên C và N cách nhau bởi dấu cách
* Dòng 2..N+1:   Ghi lần lượt các số nguyên: w[i]

**Output:**

* Một số nguyên là tổng khối lượng bò lớn nhất mà John có thể mang đi xem phim.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 259 5  81  58  42  33  61 | 242 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int dp[200][25001];

long long solution(int a[],int n, int w){

memset(dp, 0, sizeof(dp));

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

for (int j = 1; j <= w;j++){

if(j >= a[i])

dp[i][j] = max(dp[i - 1][j], dp[i-1][j - a[i]] + a[i]);

else

dp[i][j] = dp[i - 1][j];

}

}

return dp[n][w];

}

int main(){

// int t;

// cin >> t;

// while(t--){

int w, n;

cin >> w >> n;

int a[n + 1];//, v[n + 1];

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

// for (int i = 1; i <= n; i++) cin >> v[i];

cout << solution(a,n ,w) << endl;

//}

return 0;

}

10. CÁI TÚI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05027/top)

Một người có cái túi thể tích V (V<1000). Anh ta có N đồ vật cần mang theo (N≤1000), mỗi đồ vật có thể tích là A[i] (A[i]≤100) và giá trị là C[i] (C[i]≤100). Hãy xác định tổng giá trị lớn nhất của các đồ vật mà người đó có thể mang theo, sao cho tổng thể tích không vượt quá V.

**Input**

* Dòng đầu ghi số bộ test T (T<10)
* Mỗi bộ test gồm ba dòng. Dòng đầu ghi 2 số N và V. Dòng tiếp theo ghi N số của mảng A. Sau đó là một dòng ghi N số của mảng C.
* Dữ liệu vào luôn đảm bảo không có đồ vật nào có thể tích lớn hơn V.

**Output**

* Với mỗi bộ test, ghi trên một dòng giá trị lớn nhất có thể đạt được.

**Ví dụ**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int dp[2510][2510];

long long solution(int a[], int v[],int n, int w){

memset(dp, 0, sizeof(dp));

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

for (int j = 1; j <= w;j++){

if(j >= a[i])

dp[i][j] = max(dp[i - 1][j], dp[i-1][j - a[i]] + v[i]);

else

dp[i][j] = dp[i - 1][j];

}

}

return dp[n][w];

}

int main(){

int t;

cin >> t;

while(t--){

int n, w;

cin >> n >> w;

int a[n + 1], v[n + 1];

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

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

cout << solution(a, v, n ,w) << endl;

}

return 0;

}

11. GIẢI MÃ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05029/top)

Một bản tin M đã mã hóa bí mật thành các con số theo ánh xạ như sau: ‘A’->1, ‘B’->2, .., ‘Z’->26. Hãy cho biết có bao nhiêu cách khác nhau để giải mã bản tin M. Ví dụ với bản mã M=”123” nó có thể được giải mã thành ABC (1 2 3), LC (12 3), AW(1 23).

**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 xâu ký tự số M.
* T, M thỏa mãn ràng buộc: 1≤T≤100;  1≤length(M)≤40.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 123 2563 | 3 2 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

string s;cin>>s;

if(s[0]=='0'){

cout<<"0"<<endl;continue;

}

int n=s.size();

int F[n+5];

F[0]=1;F[1]=1;

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

F[i]=0;

if(s[i-1]>'0') F[i]=F[i-1];

if(s[i-2]=='1'||(s[i-2]=='2'&&s[i-1]<'7')){

//so chay tu 1-26

F[i]+=F[i-2];

}

}

cout<<F[n]<<endl;

}

return 0;

}

12. XÂU CON ĐỐI XỨNG DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05018/top)

Cho xâu S chỉ bao gồm các ký tự viết thường và dài không quá 1000 ký tự.

Hãy tìm xâu con đối xứng dài nhất của S.

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 10).
* Mỗi test gồm một xâu S có độ dài không vượt quá 1000, chỉ gồm các kí tự thường.

**Output:**Với mỗi test, in ra đáp án tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  abcbadd  aaaaa | 5  5 |
|  |  |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

string s;

while(t--){

cin>>s;

int dem=1;

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

int u=i,v=i;

while(u>=0&&v<s.size()){

if(s[u]==s[v]){

dem=max(dem,v-u+1);

u--;

v++;

}

else break;

}

}

for(int i=0;i<s.size()-1;i++){

int u=i,v=i+1;

while(u>=0&&v<s.size()){

if(s[u]==s[v]){

dem=max(dem,v-u+1);

u--;v++;

}

else break;

}

}

cout<<dem<<endl;

}

return 0;

}

13. HÌNH VUÔNG LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05019/top)

Cho một bảng số N hàng, M cột chỉ gồm 0 và 1. Bạn hãy tìm hình vuông có kích thước lớn nhất, sao cho các số trong hình vuông toàn là số 1.

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 10).
* Mỗi test bắt đầu bởi 2 số nguyên N, M (1 ≤ N, M ≤ 500).
* N dòng tiếp theo, mỗi dòng gồm M số mô tả một hàng của bảng.

**Output:**

* Với mỗi test, in ra đáp án là kích thước của hình vuông lớn nhất tìm được trên một dòng.

**Ví dụ:**

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

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int a[501][501];

int F[501][501];

int n,m;

int slove(){

int res = 0;

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

for (int j = 1; j < m;j++){

if(a[i][j] == 0)

continue;

if (a[i - 1][j - 1] == 1 && a[i][j - 1] == 1 && a[i - 1][j] == 1){

F[i][j] = min(F[i - 1][j - 1], min(F[i - 1][j], F[i][j - 1])) + 1;

}

res = max(res, F[i][j]);

}

}

return res;

}

int main(){

int t;cin >> t;

while(t--){

cin >> n >> m;

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

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

cin >> a[i][j];

F[i][j] = a[i][j];

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

F[i][j] = 1;

}

}

cout << slove() << endl;

}

}

14. ĐƯỜNG ĐI NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05020/top)

Cho bảng A[] kích thước N x M (N hàng, M cột). Bạn được phép đi xuống dưới, đi sang phải và đi xuống ô chéo dưới. Khi đi qua ô (i, j), điểm nhận được bằng A[i][j].

Hãy tìm đường đi từ ô (1, 1) tới ô (N, M) sao cho tổng điểm là nhỏ nhất.

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test gồm số nguyên dương N và M.
* N dòng tiếp theo, mỗi dòng gồm M số nguyên A[i][j] (0 ≤ A[i] ≤ 1000).

**Output:**

* Với mỗi test, in ra độ dài dãy con tăng dài nhất trên một dòng.

**Ví dụ:**

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

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n,m,a[1005][1005];

int main() {

int t; cin >> t;

while (t--) {

cin >> n >> m;

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

for (int j = 1; j <= m; j++)

cin >> a[i][j];

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

for (int j = 1; j <= m; j++) {

if (i == 1 && j != 1)

a[i][j] += a[i][j - 1];

else if (i != 1 && j == 1)

a[i][j] += a[i - 1][j];

else if (i != 1 && j != 1)

a[i][j] += min(a[i - 1][j - 1], min(a[i][j - 1], a[i - 1][j]));

}

}

cout << a[n][m] << endl;

}

return 0;

}

16. TÍNH P(N,K)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05015/top)

P(n, k) là số phép biểu diễn các tập con có thứ tự gồm k phần tử của tập gồm n phần tử. Số P(n, k) được định nghĩa theo công thức sau:

Cho số hai số n, k. Hãy tìm P(n,k) theo modulo 109+7.

**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 cặp số n, k được viết trên một dòng.
* T, n, k thỏa mãn ràng buộc: 1 ≤ T ≤ 100;  1 ≤ n,k ≤ 1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 5 2 4 2 | 20 12 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7;

int n,k;

int main(){

int t; cin >> t;

while (t--) {

cin >> n >> k;

if (n < k) cout << 0 << endl;

else {

long long p= 1;

for (int i = n; i>=n-k+1;i--){

p=(p\*i)%mod;

}

cout<<p<<endl;

}

}

return 0;

}

17. SỐ UGLY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05016/top)

Số Ugly là các số chỉ có ước số là 2, 3, 5. Theo qui ước số 1 cũng là 1 số Ugly. Dưới đây là 11 số Ugly đầu tiên: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15. Cho số tự nhiên N, nhiệm vụ của bạn là tìm số Ugly thứ N.

**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 tự nhiên N được viết trên một dòng. T, N thỏa mãn ràng buộc: 1 ≤ T ≤ 100;  1 ≤ N ≤ 104.

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 10 4 | 12 4 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

long long u[10005];

int n;

void SoUgly(){

u[1]=1;

int h = 1 , b = 1, n= 1 ;

int index = 2 ;

while (index <=10000){

long long hai = u[h]\*2 ;

long long ba = u[b]\*3 ;

long long nam = u[n]\*5;

long long ul = min(hai, min(ba,nam));

u[index] = ul ;

if (ul == hai)

h++;

if (ul == ba)

b++;

if (ul == nam)

n++;

index++;

}

}

int main(){

SoUgly();

int t ;

cin >> t;

while (t--){

cin>>n;

cout<<u[n]<<endl;

}

return 0;

}

18. DÃY CON CHUNG DÀI NHẤT CỦA BA XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05003/top)

Cho ba xâu ký tự X, Y, Z. Nhiệm vụ của bạn là tìm độ dài dãy con chung dài nhất có mặt trong cả ba xâu.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào độ dài xâu X, Y, X; dòng tiếp theo đưa vào ba xâu X, Y, Z.
* T, X, Y, Z thỏa mãn ràng buộc: 1 ≤ T ≤ 100;  1 ≤ size(X), size(Y), size(Z) ≤ 100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 8 13  geeks geeksfor geeksforgeeks  7 6 5  abcd1e2 bc12ea bd1ea | 5  3 |

  // by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int a[101][101][101];

int main() {

int t; cin >> t;

while (t--) {

int x, y, z; cin >> x >> y >> z;

string X, Y, Z; cin >> X >> Y >> Z;

memset(a, 0, sizeof(a));

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

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

for (int k = 1; k <= z; k++) {

if (X[i-1] == Y[j-1] && Y[j-1] == Z[k-1])

a[i][j][k] = a[i - 1][j - 1][k - 1]+1;

else

a[i][j][k] = max(a[i - 1][j][k], max(a[i][j - 1][k], a[i][j][k - 1]));

}

}

}

cout << a[x][y][z] << endl;

}

return 0;

}

19. TỔNG LỚN NHẤT CỦA DÃY CON TĂNG DẦN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05006/top)

Cho dãy số A[] gồm N số. Nhiệm vụ của bạn là tìm tổng lớn nhất của dãy con được sắp theo thứ tự tăng dần của dãy A[]. Ví dụ với dãy A[] = {1, 101, 2, 3, 100, 4, 5} ta có kết quả là 106 = 1 + 2 + 3 + 100. Với dãy A[] = {10, 7, 5} ta có kết quả là 10. Với dãy A[] = {1, 2, 3, 5} ta có kết quả là 11.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là số phần tử của dãy A[]; dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤103; 0≤A[i] ≤103.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  7  1 101 2 3 100 4 5  3  10 7 5  4  1 2 3 5 | 106  10  11 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n,a[1000],b[1000],summax;

int main() {

int t; cin >> t;

while (t--) {

cin >> n;

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

cin >> a[i];

b[i] = a[i];

}

summax = 0;

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

int tmp = 0;

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

if (tmp < b[j] && a[i] > a[j]) tmp = b[j];

b[i] += tmp;

if (summax < b[i]) summax = b[i];

}

cout << summax << endl;

}

return 0;

}

20. DÃY SỐ BI-TONIC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05017/top)

Một dãy số được gọi là Bi-tonic nếu nó được chia thành hai dãy đầu tăng dần và dãy tiếp theo giảm dần. Nhiệm vụ của bạn là tìm tổng lớn nhất dãy con Bi-tonic của dãy số A[]. Ví dụ với dãy A[] = {1, 15, 51, 45, 33, 100, 12, 18, 9} ta có kết quả là 194 tương ứng với dãy Bi-tonic {1, 15, 51, 100, 18, 9}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là số phần tử của dãy A[]; dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤100; 0≤A[i] ≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 6 80 60 30 40 20 10 9 1 15 51 45 33 100 12 18 9 | 210 194 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n,a[105],b[105],c[105];

int main() {

int t; cin >> t;

while (t--) {

cin >> n;

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

cin >> a[i];

b[i] = a[i];

c[i] = a[i];

}

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

int tmp1 = 0;

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

if (tmp1 < b[j] && a[i] > a[j]) tmp1 = b[j];

b[i] += tmp1;

}

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

int tmp2 = 0;

int tmp3 = 0;

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

if (tmp2<c[j] && a[i]>a[j]) {

tmp2 = c[j];

tmp3 = max(tmp3, tmp2);

}

}

c[i] += tmp2;

b[i] += tmp3;

}

int biTonic = 0;

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

if (biTonic < b[i]) biTonic = b[i];

}

cout << biTonic << endl;

}

return 0;

}

23. TỔNG BẰNG K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05024/top)

Cho một mảng A[] gồm N số nguyên và số K. Tính số cách lấy tổng các phần tử của A[] để bằng K. Phép lấy lặp các phần tử hoặc sắp đặt lại các phần tử được chấp thuận. Ví dụ với mảng A[] = {1, 5, 6}, K = 7 ta có 6 cách sau:

7 = 1 + 1 + 1+1 + 1 + 1+1 (lặp số 1 7 lần)

7 = 1 + 1 + 5 (lặp số 1)

7 = 1 + 5 + 1 (lặp và sắp đặt lại số 1)

7 = 1 + 6

7 = 6 + 1

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào N và K; dòng tiếp theo đưa vào N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤1000; 1≤A[i]≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Khi kết quả quá lớn đưa ra kết quả dưới dạng modulo với 109+7.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 3 7 1 5 6 4 14 12 3 1 9 | 6 150 |

 #include <bits/stdc++.h>

using namespace std;

long long n,k,sum,a[1001],b[1001],mod = 1e9 + 7;

int main(){

int t; cin >> t;

while (t--) {

cin >> n >> k;

sum = 0;

for (int i = 0; i < n; i++) cin >> a[i];

memset(b, 0, sizeof(b));

b[0] = 1;

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

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

if (a[j] <= i){

b[i] += b[i-a[j]];

b[i] %= mod;

}

}

}

cout << b[k] << endl;

}

return 0;

}

25. BIẾN ĐỔI XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05028/top)

Cho hai xâu ký tự str1, str2 bao gồm các ký tự in thường và các thao tác dưới đây:

* **Insert:** chèn một ký tự bất kỳ vào str1.
* **Delete:** loại bỏ một ký tự bất kỳ trong str1.
* **Replace:** thay một ký tự bất kỳ trong str1.

Nhiệm vụ của bạn là đếm số các phép Insert, Delete, Replace ít nhất thực hiện trên str1 để trở thành str2.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ đôi hai xâu str1 và str2.
* T, str1, str2 thỏa mãn ràng buộc: 1≤T≤100;  1≤length(str1),length(str2) ≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  geek gesek | 1 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int T;

cin>>T;

while(T--){

string s1, s2;

cin>>s1>>s2;

int x1=s1.size(), x2=s2.size();

int res[x1+1][x2+1];

for(int i=0; i<=x1; i++){

for(int j=0; j<=x2; j++){

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

res[i][j]=i+j;

else if(s1[i-1]==s2[j-1])

res[i][j]=res[i-1][j-1];

else

res[i][j]=min(res[i-1][j-1], min(res[i-1][j],res[i][j-1]))+1;

}

}

cout<<res[x1][x2]<<endl;

}

return 0;

}

26. TỔNG BÌNH PHƯƠNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05031/top)

Mọi số nguyên dương N đều có thể phân tích thành tổng các bình phương của các số nhỏ hơn N. Ví dụ số 100 = 102 hoặc 100 = 52 + 52 + 52 + 52. Cho số nguyên dương N. Nhiệm vụ của bạn là tìm số lượng ít nhất các số nhỏ hơn N mà có tổng bình phương bằng N.

**Input:**

* Những dòng kế tiếp đưa vào các bộ test. Mỗi test là một số tự nhiên N được viết trên 1 dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤10000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 100 6 25 | 1 3 1 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n;

int main() {

int t; cin >> t;

while (t--) {

cin >> n;

int a[n+5];

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

a[i] = i;

for (int i=4;i<=n;i++){

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

a[i]=min(a[i],a[i-j\*j]+1);

}

cout << a[n] << endl;

}

return 0;

}

27. NHÀ KHÔNG KỀ NHAU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05034/top)

Có N ngôi nhà trên một dãy phố, mỗi ngôi nhà chứa đựng một số lượng tài sản khác nhau. Một tên trộm muốn ăp cắp được nhiều nhất tài sản của dãy phố nhưng không muốn lấy tài sản của hai nhà kề nhau. Hãy cho biết, bằng cách đó tên trộm có thể đánh cắp được nhiều nhất bao nhiêu tài sản.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là một số N là số lượng ngôi nhà; dòng tiếp theo đưa vào N số là tài sản tương ứng trong mỗi ngôi nhà; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤106; 1≤A[i] ≤107.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 6 5 5 10 100 10 5 4 3 2 7 10 | 110 13 |

// hung nguyen

// nha ko ke nhau

#include<bits/stdc++.h>

using namespace std;

int n,ans;

void check(){

vector <long long> a(n);

for(int i=0;i<=n-1;i++) cin >> a[i];

long long F[n+5]={0};

F[0]=a[0];

F[1]=a[1];

F[2]=a[0]+a[2];

for(int i=3;i<=n-1;i++) F[i] = a[i] + max(F[i-2],F[i-3]);

ans=0;

ans = max(F[n-1],F[n-2]);

cout<<ans<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

check();

}

return 0;

}

28. XÂU ĐỐI XỨNG 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05032/top)

Cho xâu ký tự str. Nhiệm vụ của bạn là tìm số phép chèn tối thiểu các ký tự vào str để str trở thành xâu đối xứng. Ví dụ: str =”ab” ta có số phép chèn tối thiểu là 1 để trở thành xâu đối xứng “aba” hoặc “bab”. Với xâu str=”aa” thì số phép chèn tối thiểu là 0. Với xâu str=”abcd” có số phép chèn tối thiểu là 3 để trở thành xâu “dcbabcd”

**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 xâu ký tự được viết trên một dòng
* T, str thỏa mãn ràng buộc: 1≤T≤100;  1≤length(str)≤40.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 abcd aba geeks | 3 0  3 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

string Reverse(string s){

string t="";

for(int i=s.size()-1; i>=0; i--)

t+=s[i];

return t;

}

int slove(string s1, string s2){

int x1=s1.size(), x2=s2.size();

int a[x1+1][x2+1];

memset(a, 0, sizeof(a));

for(int i=1; i<=x1; i++){

for(int j=1; j<=x2; j++){

if(s1[i-1]==s2[j-1]) a[i][j]=a[i-1][j-1]+1;

else a[i][j]=max(a[i][j-1], a[i-1][j]);

}

}

return a[x1][x2];

}

int main(){

int t;

cin>>t;

while(t--){

string s;

cin>>s;

cout<<s.size()-slove(s, Reverse(s))<<endl;

}

return 0;

}

29. XÂU ĐỐI XỨNG 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05033/top)

Cho xâu ký tự S. Nhiệm vụ của bạn là tìm số phép loại bỏ ít nhất các ký tự trong S để S trở thành xâu đối xứng. Chú ý, phép loại bỏ phải bảo toàn tính trước sau của các ký tự trong S.

**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 xâu ký tự được viết trên một dòng
* T, str thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 aebcbda geeksforgeeks | 2 8 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

string s;

int slove(){

int n = s.size();

int X[n][n];

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

X[i][i] = 1;

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

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

int j =i+k-1;

if (s[i]==s[j]&&k==2)

X[i][j] = 2;

else if (s[i]==s[j])

X[i][j]=X[i+1][j-1]+2;

else

X[i][j]=max(X[i][j-1],X[i+1][j]);

}

}

return n - X[0][n - 1];

}

int main(){

int t;

cin >> t;

while(t--){

cin >> s;

cout << slove() << endl;

}

return 0;

}

30. DÃY CON DÀI NHẤT CÓ TỔNG CHIA HẾT CHO K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05010/top)

Cho một dãy gồm n ( n ≤ 1000) số nguyên dương A1, A2, ..., An và số nguyên dương k (k ≤ 50). Hãy tìm dãy con gồm nhiều phần tử nhất của dãy đã cho sao cho tổng các phần tử của dãy con này chia hết cho k.

**Input:**Dòng đầu ghi số bộ test T (T<10). Mỗi bộ test gồm 2 dòng. Dòng đầu tiên chứa hai số n, k.  Dòng tiếp theo ghi n số của dãy A. Các số đều không vượt quá 100.

**Output:**Gồm 1 dòng duy nhất ghi số lượng phần tử của dãy con dài nhất thoả mãn. Dữ liệu vào luôn đảm bảo sẽ có ít nhất một dãy con có tổng chia hết cho k.

**Ví dụ:**

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

// by hung nguyen

// day con chia het cho k

#include<bits/stdc++.h>

using namespace std;

const int tmp = -1e5;

int n,k;

void xuly(){

cin >> n >> k;

int F[n+5][k]={0};

F[0][0]=0;

for(int i=1;i<=k-1;i++) F[0][i]=tmp;

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

int x;cin >> x;

x%=k;

for(int j=0;j<=k-1;j++) F[i][j] = max(F[i-1][j],F[i-1][(j+k-x)%k]+1);

}

cout<<F[n][0]<<endl;;

}

int main(){

int t;

cin >> t;

while(t--){

xuly();

}

return 0;

}

33. DI CHUYỂN VỀ GỐC TỌA ĐỘ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA05030/top)

Giả sử bạn đang ở điểm có tọa độ nguyên dương (n,m) và cần dịch chuyển về tọa độ (0,0). Mỗi bước dịch chuyển bạn chỉ được phép dịch chuyển đến tọa độ (n-1, m) hoặc (n, m-1); Từ ô (0,m), hoặc (n, 0) thì có thể di chuyển 1 bước để về gốc (0,0).

Hãy đếm số cách bạn có thể dịch chuyển về tọa độ (0,0).

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ n, m được viết cách nhau một vài khoảng trống.
* T, n, m thỏa mãn ràng buộc: 1≤T≤100;  1≤n, m ≤25.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 3 2 3 6 3 0 | 10 84 1 |

// hung nguyen

// di chuyen ve goc toa do

#include<bits/stdc++.h>

using namespace std;

int n,m;

long long F[40][40];

void slove(){

// khoi tao cac phan tu f=0

memset(F,0,sizeof(F));

for(int i=0;i<=25;i++) F[i][0]=1;

for(int j=0;j<=25;j++)F[0][j]=1;

//khoi tao gia tri ban dau cho F[0][0]

F[0][0]=0;

for(int i=0;i<=25;i++){

for(int j=0;j<=25;j++){

if(i==0 || j==0) continue;

F[i][j] = F[i-1][j] + F[i][j-1];

}

}

}

int main(){

int t;

cin >> t;

slove();

while(t--){

cin>>n>>m;

cout<<F[m][n]<<endl;

}

return 0;

}

Dạng 6:Sắp xếp – tìm kiếm

1.SẮP XẾP ĐỔI CHỖ TRỰC TIẾP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06023/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. **Dữ liệu vào:**Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100). **Kết quả:** 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<bits/stdc++.h>

using namespace std;

int a[101],n;

void in(){

for(int i=0;i<n;i++) cout<<a[i]<<" ";

cout<<endl;

}

void interchangeSort(){

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

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

if(a[i]>a[j]) swap(a[i],a[j]);

}

cout<<"Buoc "<<i+1<<":"<<" ";

in();

}

}

int main(){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

interchangeSort();

}

2. SẮP XẾP CHỌN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06024/top)

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

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

**Kết quả:** 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<bits/stdc++.h>

using namespace std;

int a[101],n;

void in(){

for(int i=0;i<n;i++) cout<<a[i]<<" ";

cout<<endl;

}

void SelectionSort(){

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

int min=i;

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

if(a[j]<a[min])

min=j;

}

swap(a[i],a[min]);//doi cho min voi phan tu dau

cout<<"Buoc "<<i+1<<":"<<" ";

in();

}

}

int main(){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

SelectionSort();

}

3. SẮP XẾP CHÈN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06025/top)

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

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

**Kết quả:** 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<bits/stdc++.h>

using namespace std;

int a[101],n;

void in(int i){

for(int j=0;j<=i;j++) cout<<a[j]<<" ";

cout<<endl;

}

void insertionSort(){

cout<<"Buoc 0: "<<a[0]<<endl;;

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

int tmp=a[i];

int j=i-1;

while(j>=0&&a[j]>tmp){

a[j+1]=a[j];

j--;

}

a[j+1]=tmp;

cout<<"Buoc "<<i<<":"<<" ";

in(i);

}

}

int main(){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

insertionSort();

}

4. SẮP XẾP NỔI BỌT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06026/top)

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

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

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

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

int a[101],n;

void in(){

for(int j=0;j<n;j++) cout<<a[j]<<" ";

cout<<endl;

}

void BubbleSort(){

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

int ok=0;

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

if(a[j]>a[j+1]){

swap(a[j],a[j+1]);

ok=1;

}

}

if(ok==0) break;

cout<<"Buoc "<<i+1<<":"<<" ";

in();

}

}

int main(){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

BubbleSort();

}

5. SẮP XẾP XEN KẼ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06001/top)

Cho mảng A[] gồm n số nguyên khác nhau. Hãy đưa ra các phần tử của mảng theo khuôn dạng lớn nhất, nhỏ nhất, lớn thứ hai, nhỏ thứ 2, … Ví dụ với A[] = {9, 7, 12, 8, 6, 5} ta đưa ra : 12, 5, 9, 6, 8, 7.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số A [i] của mảng A [];các số được viết cách nhau một vài khoảng trống.
* T, n thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

for(int i=0;i<n/2;i++){

cout<<a[n-1-i]<<" "<<a[i]<<" ";

}

if(n%2==1) cout<<a[n/2];

cout<<endl;

}

return 0;

}

6. SẮP XẾP THEO GIÁ TRỊ TUYỆT ĐỐI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06002/top)

Cho mảng A[] gồm n phần tử và số X. Hãy đưa sắp xếp các phần tử của mảng theo trị tuyệt đối của |X - A[i] |. Ví dụ với A[] = {10, 5, 3, 9, 2} và X = 7 ta đưa ra mảng được sắp xếp theo nguyên tắc kể trên: A[] = {5, 9, 10, 3, 2} vì |7-10|=3, |7-5|=2, |7-3|=4, |7-9|=2, |7-2|=5.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và X; dòng tiếp theo là n số A [i] của mảng A [];các số được viết cách nhau một vài khoảng trống.
* T, n, X thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, X, A[i] ≤105.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,k,a[2005];

bool comp(pair<int,int> a,pair<int,int> b){

if(abs(a.first-k)!=abs(b.first-k)){

return abs(a.first-k)<abs(b.first-k);

}

else return a.second<b.second;

}

void in(){

cin>>n>>k;

vector<pair<int,int>> v(n);

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

int x;

cin>>x;

v[i]=make\_pair(x,i);//so va chi so

}

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

for(pair<int,int>p:v){

cout<<p.first<<" ";

}

cout<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

in();

}

return 0;

}

7. ĐỔI CHỖ ÍT NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06003/top)

Cho mảng A[] gồm n phần tử. Hãy tìm số phép đổi chỗ ít nhất giữa các phần tử của mảng để mảng A[] được sắp xếp. Ví dụ với A[] = {4, 3, 2, 1} ta cần thực hiện ít nhất 2 phép đổi chỗ: Swap(A[0], A[3]),  Swap(A[1], A[2]).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và X; dòng tiếp theo là n số A [i] của mảng A [];các số được viết cách nhau một vài khoảng trống.
* T, n thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[1001];

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

int res=0;

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

//lay min tu vi tri i ve cuoi, neu tmp!=i doi cho

int tmp=min\_element(a+i,a+n)-a;

if(tmp>i){

swap(a[i],a[tmp]);res++;

}

}

cout<<res<<endl;

}

return 0;

}

8. HỢP VÀ GIAO CỦA HAI DÃY SỐ 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06004/top)

Cho mảng A[] gồm n phần tử, mảng B[] gồm m phần tử khác nhau. Các phần tử của mảng A[] và B[] đã được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa A[] và B[]. Ví dụ với A[] = {1, 3, 4, 5, 7}, B[]={2, 3, 5, 6} ta có mảng hợp Union = {1, 2, 3, 4, 5, 6, 7}, mảng giao Intersection = {3, 5}. In ra đáp án theo giá trị phần tử từ nhỏ đến lớn.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m là số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A [i] của mảng A [];dòng tiếp theo là m số B[i] của mảng B[]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, A[i], B[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m, A[i], B[i] ≤105.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,m;cin>>n>>m;

int a[n+m];

map<int,int> ma;

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

cin>>a[i];

ma[a[i]]++;//dem

}

for(auto&p:ma){//tuong tu vong for

cout<<p.first<<" ";//in ra ca phan tu da dc sx

}

cout<<endl;

for(auto&p:ma){

if(p.second!=1) cout<<p.first<<" ";//in cac phan tu xuat hien lon hon 1 lan

}

cout<<endl;

}

return 0;

}

9.  HỢP VÀ GIAO CỦA HAI DÃY SỐ 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06005/top)

Cho mảng A[] gồm n phần tử, mảng B[] gồm m phần tử khác nhau. Các phần tử của mảng A[] và B[] chưa được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa A[] và B[]. Ví dụ với A[] = {7, 1, 5, 2, 3, 6}, B[]={3, 8, 6, 20, 7} ta có mảng hợp Union = {1, 2, 3, 5, 6, 7, 8, 20}, mảng giao Intersection = {3, 6}.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m là số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A [i] của mảng A [];dòng tiếp theo là m số B[i] của mảng B[]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, A[i], B[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m, A[i], B[i] ≤105.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 1 6 5 7 1 5 2 3 6 3 8 6 20 7 | 1 2 3 5 6 7 8 20  2 6 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,m;cin>>n>>m;

int a[n+m];

map<int,int> ma;

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

cin>>a[i];

ma[a[i]]++;

}

for(auto&p:ma){

cout<<p.first<<" ";

}

cout<<endl;

for(auto&p:ma){

if(p.second!=1) cout<<p.first<<" ";

}

cout<<endl;

}

return 0;

}

10.  SẮP XẾP [0 1 2]

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06006/top)

Cho mảng A[] gồm n phần tử. Các phần tử của mảng A[] chỉ bao gồm các số 0, 1, 2. Hãy sắp xếp mảng A[] theo thứ tự tăng dần. Ví dụ với A[] = {0, 2, 1, 2, 0} ta kết quả A[] = {0, 0, 1, 2, 2}.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A [i] của mảng A []các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 0≤ A[i] ≤2; 1≤ n ≤106.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 0 2 1 2 0  3 0 1 0 | 0 0 1 2 2  0 1 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

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

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

}

cout<<endl;

}

return 0;

}

11. SẮP XẾP DÃY CON LIÊN TỤC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06007/top)

Cho mảng A[] gồm n phần tử. Hãy tìm dãy con liên tục của mảng A[R], .., A[L] sao cho khi sắp xếp lại dãy con ta nhận được một mảng được sắp xếp. Ví dụ với A[] = {10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60} ta chỉ cần sắp xếp lại dãy con từ A[4],.., A[9]: {30, 25, 40, 32, 31, 35} để có mảng được sắp.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A [i] của mảng A []các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤106; 0≤ A[i] ≤107.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 11 10 12 20 30 25 40 32 31 35 50 60 9 0 1 15 25 6 7 30 40 50 | 4 9 3 6 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n],b[n];

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

cin>>a[i];b[i]=a[i];

}

sort(a,a+n);

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

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

cout<<i+1<<" ";break;

}

}

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

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

cout<<i+1;break;

}

}

cout<<endl;

}

return 0;

}

12. ĐẾM CẶP

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06008/top)

Cho mảng X[] gồm n phần tử và mảng Y[] gồm m phần tử. Hãy đếm số các cặp xy>yx, trong đó x€X[] và y€Y[]. Ví dụ X[] = {2, 1, 6 }, Y[] = {1, 5} ta có kết quả là 3 cặp (2, 1), (2, 5), (6, 1).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng X[] và Y[]; dòng tiếp theo là n số X[i] của mảng X[]; dòng cuối cùng là m số của mảng Y[]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, X[i], Y[j] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m ≤105; 0≤ X[i], Y[j] ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,m,a[100001],b[100001];

long long slove(int dem[],int x){

if(x==0) return 0;

if(x==1) return dem[0];

int y=upper\_bound(b,b+m,x)-b;

long long tmp=m-y;

tmp+=dem[0]+dem[1];

if(x==2) tmp-=(dem[3]+dem[4]);

if(x==3) tmp+=dem[2];

return tmp;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n>>m;

for(int i=0;i<n;i++) cin>>a[i];

int dem[10]={0};

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

cin>>b[i];

if(b[i]<5) dem[b[i]]++;

}

sort(b,b+m);

long long res=0;

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

res+=slove(dem,a[i]);

}

cout<<res<<endl;

}

return 0;

}

13.  CẶP SỐ TỔNG BẰNG K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06009/top)

Cho mảng A[] gồm n phần tử và số k. Đếm tất cả các cặp phần tử của mảng có tổng bằng k. Ví dụ A[] = {1, 5, 3, 4, 2 }, k = 7 ta có kết quả là 2 cặp (3, 4), (5, 2).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[] và k; dòng tiếp theo là n số A[i] của mảng A[]các số được viết cách nhau một vài khoảng trống.
* T, n, k, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤100; 0≤ k ≤100, 0≤ A[i] ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 9  1 5 4 1 2 3 2  1 1 1 | 1 3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int c[100][100];

int tohop(int n){

for (int i=0;i<=2;i++)

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

if (i==0||i==j) c[i][j]=1;

else c[i][j]=c[i][j-1]+c[i-1][j-1];

return c[2][n];

}

//vd 8=4+4 4 4 4 4 4 ta lay 2 cap bat ky

int main(){

int t;cin>>t;

while(t--){

int n,k;cin>>n>>k;

int a[n];

map<int,int> m;

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

cin>>a[i];

m[a[i]]++;

}

sort(a,a+n);

int dem=0;

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

if(k-a[i]==a[i]){

dem+=tohop(m[a[i]]);

m[a[i]]=1;

}

else{

if(binary\_search(a+i,a+n,k-a[i])) dem+=m[k-a[i]];//xet tu a+i den cuoi cong so lan xuat hien

//vd 9=5+4 5 4 4 4 4 a[i]=5 k-a[i]=4 ->+so lan xuat hien cua 4

}

}

cout<<dem<<endl;

}

return 0;

}

13.0.CẶP SỐ TỔNG BẰNG K

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06009/top)

Cho mảng A[] gồm n phần tử và số k. Đếm tất cả các cặp phần tử của mảng có tổng bằng k. Ví dụ A[] = {1, 5, 3, 4, 2 }, k = 7 ta có kết quả là 2 cặp (3, 4), (5, 2).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[] và k; dòng tiếp theo là n số A[i] của mảng A[]các số được viết cách nhau một vài khoảng trống.
* T, n, k, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤100; 0≤ k ≤100, 0≤ A[i] ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 9  1 5 4 1 2 3 2  1 1 1 | 1 3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int c[100][100];

int tohop(int n){

for (int i=0;i<=2;i++)

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

if (i==0||i==j) c[i][j]=1;

else c[i][j]=c[i][j-1]+c[i-1][j-1];

return c[2][n];

}

//vd 8=4+4 4 4 4 4 4 ta lay 2 cap bat ky

int main(){

int t;cin>>t;

while(t--){

int n,k;cin>>n>>k;

int a[n];

map<int,int> m;

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

cin>>a[i];

m[a[i]]++;

}

sort(a,a+n);

int dem=0;

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

if(k-a[i]==a[i]){

dem+=tohop(m[a[i]]);

m[a[i]]=1;

}

else{

if(binary\_search(a+i,a+n,k-a[i])) dem+=m[k-a[i]];//xet tu a+i den cuoi cong so lan xuat hien

//vd 9=5+4 5 4 4 4 4 a[i]=5 k-a[i]=4 ->+so lan xuat hien cua 4

}

}

cout<<dem<<endl;

}

return 0;

}

14.  SẮP XẾP CHỮ SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06010/top)

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là đưa ra mảng đã được sắp xếp bao gồm các chữ số của mỗi phần tử trong A[]. Ví dụ A[] = {110, 111, 112, 113, 114 }ta có kết quả là {0, 1, 2, 3, 4}.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A[i] ; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤107; 0≤ A[i] ≤1016.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 3 131 11 48 4 111 222 333 446 | 1 3 4 8 1 2 3 4 6 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

long long a[n];

map<int,int> m;

int dem=0;

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

cin>>a[i];

int ok=0;

if(a[i]==0&&ok==0){

m[0]=1;

dem++;ok=1;

}

while(a[i]!=0){

long long r=a[i]%10;

m[r]++;

if(m[r]==1) dem++;

a[i]/=10;

}

}

for(int i=0;i<=9;i++){

if(m[i]!=0) cout<<i<<" ";

}

cout<<endl;

}

return 0;

}

15. TỔNG GẦN 0 NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06011/top)

Cho mảng A[] gồm n phần tử, hãy tìm cặp phần tử có tổng gần nhất so với 0.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[]; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤T≤100; 2≤N ≤103, -106≤A[i] ≤106.

**Output**:

* Đưa ra tổng gần nhất với 0 của cặp phần tử.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 3 -8 -66 -60  6 -21 -67 -37 -18 4 -65 | -68 -14 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

// duyet trau

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n],min=1000000,ans=0;

for(int i=0;i<n;i++) cin>>a[i];

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

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

int x=abs(a[i]+a[j]);

if(min>x){

min=x;ans=a[i]+a[j];

}

}

}

cout<<ans<<endl;

}

return 0;

}

16.  PHẦN TỬ LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06012/top)

Cho mảng A[] gồm n phần tử, hãy tìm k phần tử lớn nhất của mảng. Các phần tử được đưa ra theo thứ tự giảm dần.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N và K; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤K < N ≤103, 1≤A[i] ≤106.

**Output**:

* Đưa ra k phần tử lớn nhất trên một dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 3  10 7 9 12 6  6 2  9 7 12 8 6 5 | 12 10 9  12 9 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,k;cin>>n>>k;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);

for(int i=n-1;i>=n-k;i--) cout<<a[i]<<" ";

cout<<endl;

}

return 0;

}

17. SỐ LẦN XUẤT HIỆN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06013/top)

Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy tìm số lần xuất hiện số X trong mảng. Nếu số lần xuất hiện số x trong mảng là 0 hãy đưa ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N và X; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤103, 1≤A[i], X ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,x;cin>>n>>x;

map<int,int> m;

int a[n];

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

cin>>a[i];

m[a[i]]++;

}

if(m[x]!=0) cout<<m[x]<<endl;

else cout<<"-1"<<endl;

}

return 0;

}

18. TỔNG CẶP SỐ NGUYÊN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06014/top)

Cho số tự nhiên N. Hãy tìm cặp số nguyên tố đầu tiên có tổng là N. Nếu không tồn tại cặp số nguyên tố có tổng bằng N, hãy đưa ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm là một số N được ghi trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 4  8 | 2 2  2 5 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int nt(int n){

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

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

}

return 1;

}

int main(){

int t;

cin >> t;

while(t--){

int n; cin>>n;

int ok=0;

if(n<4) cout<<"-1";

else{

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

if(nt(i)&&nt(n-i)){

cout<<i<<" "<<n-i;ok=1;break;

}

}

if(ok==0) cout<<"-1";

}

cout<<endl;

}

return 0;

}

19. MERGE SORT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06015/top)

Cho mảng A[] gồm N phần tử chưa được sắp xếp. Nhiệm vụ của bạn là sắp xếp các phần tử của mảng A[] theo thứ tự tăng dần bằng thuật toán Merge Sort.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số N tương ứng với số phần tử của mảng A[]; phần thứ 2 là N số của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N, A[i]≤106.

Output:

* Đưa ra kết quả các test theo từng dòng.

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n; cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

sort(a,a+n);//chinh la merge sort

for(int i=0;i<n;i++) cout<<a[i]<<" ";

cout<<endl;

}

return 0;

}

20. TÍCH LỚN NHẤT - NHỎ NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06016/top)

Cho mảng A[] gồm n phần tử và mảng B[] gồm m phần tử. Nhiệm vụ của bạn là tìm tích giữa phần tử lớn nhất của mảng A[] và phần tử nhỏ nhất của mảng B[]. Ví dụ A[] = {5, 7, 112, 9, 3, 6, 2 }, B[] = {1, 2, 6, -1, 0, 9} ta có kết quả là -9 = 9\*(-1).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i] ; dòng cuối cùng là m số B[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, A[i], B[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m ≤106; -108≤ A[i] ≤108.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 6 6 5 7 9 3 6 2 1 2 6 -1 0 9 6 6  1 4 2 3 10 2 4 2 6 5 2 9 | -9 20 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,m; cin>>n>>m;

long long a[n],b[m];

for(int i=0;i<n;i++) cin>>a[i];

for(int i=0;i<n;i++) cin>>b[i];

sort(a,a+n);sort(b,b+m);

cout<<a[n-1]\*b[0]<<endl;

}

return 0;

}

21. TRỘN HAI DÃY

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06017/top)

Cho mảng A[] gồm n phần tử và mảng B[] gồm m phần tử. Nhiệm vụ của bạn là hợp nhất hai mảng A[] và B[] để được một mảng mới đã được sắp xếp. Ví dụ A[] = {5, 7, 112, 9, 3, 6, 2 }, B[] = {1, 2, 6, -1, 0, 9} ta có kết quả là C[] = {-1, 1, 0, 2, 3, 5, 6, 6, 7, .

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i] ; dòng cuối cùng là m số B[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, A[i], B[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m ≤106; -108≤ A[i] ≤108.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 1  3 3  10 5 15  20 3 2 | 2 3 5 10 15 20 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,m; cin>>n>>m;

int a[n+m];

for(int i=0;i<n+m;i++) cin>>a[i];

sort(a,a+n+m);

for(int i=0;i<n+m;i++) cout<<a[i]<<" ";

cout<<endl;

}

return 0;

}

22. BỔ SUNG PHẦN TỬ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06018/top)

Cho mảng A[] gồm n số nguyên dương. Gọi L, R là max và min các phần tử của A[]. Nhiệm vụ của bạn là tìm số phần tử cần thiết cần thêm vào mảng để mảng có đầy đủ các số trong khoảng [L, R]. Ví dụ A[] = {5, 7, 9, 3, 6, 2 } ta nhận được kết quả là 2 tương ứng với các số còn thiếu là 4, 8.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n, tương ứng với số phần tử của mảng A[]; dòng tiếp theo là n số A[i] ; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, A[i] ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 4 5 3 8 6 3 2 1 3 | 1 0 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

map<int,int> m;

int a[n],dem=0;

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

cin>>a[i];

m[a[i]]=1;

}

sort(a,a+n);

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

if(m[i]!=1) dem++;

}

cout<<dem<<endl;

}

return 0;

}

23. SẮP XẾP THEO SỐ LẦN XUẤT HIỆN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06019/top)

Cho mảng A[] gồm n số nguyên. Nhiệm vụ của bạn là sắp xếp mảng theo số lần xuất hiện các phần tử của mảng. Số xuất hiện nhiều lần nhất đứng trước. Nếu hai phần tử có số lần xuất hiện như nhau, số nhỏ hơn đứng trước. Ví dụ A[] = {5, 5, 4, 6, 4 }, ta nhận được kết quả là A[] = {4, 4, 5, 5, 6}.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n, tương ứng với số phần tử của mảng A[] và số k; dòng tiếp theo là n số A[i] ; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤104; 1≤ k ≤103; 1≤ A[i] ≤105.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 5 5 4 6 4 5 9 9 9 2 5 | 4 4 5 5 6 9 9 9 2 5 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

map<int,int> m;

int a[n],b[n],dem=0,k=0;

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

cin>>a[i];

m[a[i]]++;

}

for(auto&p:m){

b[k++]=p.second;

}

sort(b,b+k);

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

for(auto&p:m){

if(b[i]==p.second){

while(p.second--) cout<<p.first<<" ";

}

}

}

cout<<endl;

}

return 0;

}

24. TÌM KIẾM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06020/top)

Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy đưa ra 1 nếu X có mặt trong mảng A[], ngược lại đưa ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n, X là số các phần tử của mảng A[] và số X cần tìm; dòng tiếp theo đưa vào n số A[i] (1≤i≤n) các số được viết cách nhau một vài khoảng trống.
* T, n, A, X thỏa mãn ràng buộc: 1≤T≤100; 1≤N, X, A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  5 16 2 4 7 9 16 7 98 1 22 37 47 54 88 96 | 1 -1 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,X;cin>>n>>X;

int a[n],dem=-1;

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

cin>>a[i];

if(a[i]==X) dem=1;

}

cout<<dem<<endl;

}

return 0;

}

25. TÌM KIẾM TRONG DÃY SẮP XẾP VÒNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06021/top)

Một mảng được sắp được chia thành hai đoạn tăng dần được gọi là mảng sắp xếp vòng. Ví dụ mảng A[] = { 5, 6, 7, 8, 9, 10, 1, 2, 3, 4} là mảng sắp xếp vòng. Cho mảng A[] gồm n phần tử, hãy tìm vị trí của phần tử x trong mảng A[] với thời gian log(n).

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n và x; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i], x  thỏa mãn ràng buộc: 1≤T≤100; 1≤N, x, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

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

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n,X;cin>>n>>X;

int a[n],dem;

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

cin>>a[i];

if(a[i]==X) dem=i+1;

}

cout<<dem<<endl;

}

return 0;

}

26. SỐ NHỎ NHẤT VÀ NHỎ THỨ HAI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06022/top)

Cho mảng A[] gồm n phần tử, hãy đưa ra số nhỏ nhất và số nhỏ thứ hai của mảng. Nếu không có số nhỏ thứ hai, hãy đưa ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[]; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤N, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

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

#include<bits/stdc++.h>

using namespace std;

bool ok[1000000+7];

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;

cin>>t;

while(t--){

long n;

cin>>n;

vector<long > a;

memset(ok,false,sizeof(ok));

int val;

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

cin>>val;

if(!ok[val]){

ok[val]=true;

a.push\_back(val);

}

}

if(a.size()<2) cout<<-1;

else{

sort(a.begin(), a.end());

cout<<a[0]<<" "<<a[1];

}

cout<<endl;

}

return 0;

}

27.  XÓA DỮ LIỆU TRONG DSLK ĐƠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_005/top)

Cho danh sách liên kết đơn lưu giữ các số nguyên được quản lý bởi con trỏ First. Viết chương trình con xóa tất cả các phần tử có giá trị bằng x trong danh sách liên kết đơn; chương trình con trả về số lượng các phần tử đã xóa. Sau khi xóa xong, liệt kê các phần tử còn lại trong danh sách liên kết đơn First. Ví dụ: Ta có Input sau:

14 : là số lượng phần tử trong danh sách

1 1 1 4 5 1 1 1 1 7 1 8 1 9   : là 14 phần tử

1 : là số cần xóa

Output :  4 5 7 8 9

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

int k;cin>>k;

for(int i=0;i<n;i++) if(a[i]!=k) cout<<a[i]<<" ";

return 0;

}

28. LỌC DỮ LIỆU TRÙNG TRONG DSLK ĐƠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/CTDL_006/top)

Cho danh sách liên kết đơn lưu giữ các số nguyên được quản lý bởi con trỏ First. Viết chương trình con lọc tất cả các phần tử có giá trị trùng nhau trong danh sách liên kết đơn First, chỉ để lại 1 phần tử đại diện cho nhóm trùng. Sau khi lọc xong, liệt kê các phần tử  trong danh sách liên kết đơn First.

Ví dụ: Ta có Input :

12 : là số lượng phân tử trong danh sách

1 1 1 4  5 1 4  7 7 8 1 9 : là số phần tử

Output : 1 4 5 7 8 9

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;cin>>n;

int a[n],dem[100001]={0};

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

cin>>a[i];dem[a[i]]=1;

}

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

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

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

dem[a[i]]=0;

}

}

return 0;

}

29. GIAO CỦA BA DÃY SỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06040/top)

Cho ba dãy số A[], B[], C[] gồm N1, N2, N3 phần tử đã được sắp xếp. Hãy đưa ra các phần tử có mặt trong cả ba dãy theo thứ tự tăng dần. Nếu không có đáp án, in ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm bốn dòng: dòng thứ nhất đưa vào N1, N2, N3 là số phần tử của mảng A[], B[], C[]; các dòng tiếp theo đưa vào 3 dãy A[], B[], C[].
* Ràng buộc: 1≤T≤100; 1≤ N1, N2, N3 ≤106, 0≤ A[i], B[j], C[k] ≤1018.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1 6 5 8 1 5 10 20 40 80 6 7 20 80 100 3 4 15 20 30 70 80 120 | 20 80 |

// by hung nguyen 2

#include <bits/stdc++.h>

using namespace std;

void in(long long n1,long long n2,long long n3,long long a[],long long b[],long long c[]){

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

bool ok=0;

while(i<n1&&j<n2&&k<n3){

if(a[i]==b[j]&&b[j]==c[k]){

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

i++;j++,k++;

ok=1;

}

else if(a[i]<b[j]) i++;

else if(b[j]<c[k]) j++;

else k++;

}

if(!ok) cout<<"-1";

}

int main(){

int t;

cin >> t;

while(t--){

long long n1,n2,n3;cin>>n1>>n2>>n3;

long long a[n1],b[n2],c[n3];

for(int i=0;i<n1;i++) cin>>a[i];

for(int i=0;i<n2;i++) cin>>b[i];

for(int i=0;i<n3;i++) cin>>c[i];

in(n1,n2,n3,a,b,c);

cout<<endl;

}

return 0;

}

30. SẮP XẾP CHẴN LẺ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA06044/top)

Cho dãy số a[] có n phần tử, đánh số từ 1 đến n. Hãy sắp xếp các phần tử ở vị trí lẻ theo thứ tự tăng dần, các phần tử ở vị trí chẵn theo thứ tự giảm dần.

**Input**

Dòng đầu tiên ghi số n, không quá 105

Dòng thứ 2 ghi n số của dãy a[] (a ≤ a[i] ≤ 109)

**Output**

Ghi ra dãy số kết quả trên một dòng

**Ví dụ**

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

// by hung nguyen 2

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

cin>>n;

long long a[n],b[n],c[n];

int h=0,k=0;

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

cin>>a[i];

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

else c[k++]=a[i];

}

sort(b,b+h,greater<long long>());

sort(c,c+k);

int max1=max(h,k);

for(int i=0;i<max1;i++){

if(i<k)cout<<c[i]<<" ";

if(i<h)cout<<b[i]<<" ";

}

return 0;

}

Dạng 7: Ngăn xếp

1. DÃY NGOẶC ĐÚNG DÀI NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07021/top)

Cho một xâu chỉ gồm các kí tự ‘(‘ và ‘)’. Một dãy ngoặc đúng được định nghĩa như sau:

-     Xâu rỗng là 1 dãy ngoặc đúng.

-     Nếu A là 1 dãy ngoặc đúng thì (A) là 1 dãy ngoặc đúng.

-     Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một xâu S. Nhiệm vụ của bạn là hãy tìm dãy ngoặc đúng dài nhất xuất hiện trong xâu đã cho.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm một xâu S có độ dài không vượt quá 105 kí tự.

**Output:**Với mỗi test in ra một số nguyên là độ dài dãy ngoặc đúng dài nhất tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  ((()  )()())  ()(())))) | 2  4  6 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;cin>>t;

while(t--){

string s;cin>>s;

stack<int> n;

n.push(-1);

int res=0;

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

if(s[i]=='(') n.push(i);

else{

n.pop();

if(n.size()>0) res=max(res,i-n.top());

else n.push(i);

}

}

cout<<res<<endl;

}

return 0;

}

2. NGĂN XẾP 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07001/top)

Cho một ngăn xếp các số nguyên. Các thao tác gồm 3 lệnh: push, pop và show. Trong đó thao tác push kèm theo một giá trị cần thêm (không quá 1000). Hãy viết chương trình ghi ra kết quả của các lệnh show.

**Input:** Gồm nhiều dòng, mỗi dòng chứa một lệnh push, pop hoặc show. Input đảm bảo số lượng phần tử trong stack khi nhiều nhất cũng không vượt quá 200.

**Output:** Ghi ra màn hình các phần tử đang có trong stack theo thứ tự lưu trữ mỗi khi gặp lệnh show. Các số viết cách nhau đúng một khoảng trống. Nếu trong stack không còn gì thì in ra dòng “empty”

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| push 3  push 5  show  push 7  show  pop  pop  show | 3 5  3 5 7  3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

string s;

stack<int> a;

while(cin>>s){

if(s=="push"){

int n;cin>>n;

a.push(n);

}

else if(s=="pop") a.pop();

else if(s=="show"){

if(a.size()==0){

cout<<"empty\n";

continue;

}

else{

vector<int> v;

while(!a.empty()){

v.push\_back(a.top());

a.pop();

}

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

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

cout<<v[i]<<" ";

a.push(v[i]);

}

cout<<endl;

}

}

}

return 0;

}

3. NGĂN XẾP 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07002/top)

Yêu cầu bạn xây dựng một stack với các truy vấn sau đây:

“PUSH x”: Thêm phần tử x vào stack (0 ≤ x ≤ 1000).

“PRINT”: In ra phần tử đầu tiên của stack. Nếu stack rỗng, in ra “NONE”.

“POP”: Xóa phần tử đầu tiên của stack. Nếu stack rỗng, không làm gì cả.

**Input:**

Dòng đầu tiên là số lượng truy vấn Q (Q ≤ 100000).

Mỗi truy vấn có dạng như trên.

**Output:**

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của stack. Nếu stack rỗng, in ra “NONE”.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 9  PUSH 1  PUSH 2  POP  PRINT  PUSH 3  PRINT  POP  POP  PRINT | 1  3  NONE |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

string s;

stack<int> a;

while(t--){

cin>>s;

if(s=="PUSH"){

int n;cin>>n;

a.push(n);

}

else if(s=="POP"&&a.size()>0) a.pop();

else if(s=="PRINT"){

if(a.size()==0){

cout<<"NONE\n";

}

else{

cout<<a.top()<<endl;

}

}

}

return 0;

}

4.  KIỂM TRA BIỂU THỨC SỐ HỌC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07003/top)

Cho biểu thức số học, hãy cho biết biểu thức số học có dư thừa các cặp ký hiệu ‘(’,’) ‘ hay không?

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  ((a+b))  (a + (b)/c)  (a + b\*(c-d)) | Yes  Yes  No |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

cin.ignore();

while(t--){

string s;

getline(cin,s);

stack<char> a;

bool ok;

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

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

ok=true;

char top=a.top();

a.pop();

while(top!='('){

if(top=='+'||top=='-'||top=='\*'||top=='/'){

ok=false;

}

top=a.top();a.pop();

}

if(ok) break;

}

else a.push(s[i]);

}

if(ok) cout<<"Yes"<<endl;

else cout<<"No"<<endl;

}

return 0;

}

5. ĐẾM SỐ DẤU NGOẶC ĐỔI CHIỀU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07004/top)

Cho một xâu chỉ gồm các kí tự ‘(‘, ‘) và có độ dài chẵn. Hãy đếm số lượng dấu ngoặc cần phải đổi chiều ít nhất, sao cho xâu mới thu được là một dãy ngoặc đúng.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 1 xâu S có độ dài không vượt quá 100 000, chỉ gồm dấu ( và ).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 4  ))((  ((((  (((())  )(())((( | 2  2  1  3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

string s;

cin>>s;

//xay dung stack luu lai cac ngoac ko tm

stack<char> a;

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

if(s[i]=='('||(s[i]==')'&&a.size()==0)) a.push(s[i]);

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

//neu co ngoac tm thi pop ra khoi ngan xep

if(a.size()>0&&a.top()=='(') a.pop();

else a.push(s[i]);

}

}

int dem1=0,dem2=0;

for(int i=a.size();i>0;i--){

if(a.top()=='(') dem1++;

else dem2++;

a.pop();

}

int res=dem1/2+dem2/2+dem1%2+dem2%2;

cout<<res<<endl;

}

return 0;

}

6. BIẾN ĐỔI TIỀN TỐ - TRUNG TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07009/top)

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng trung tố.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  \*+AB-CD  \*-A/BC-/AKL | ((A+B)\*(C-D))  ((A-(B/C))\*((A/K)-L) |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

// y tuong duyet tu cuoi ve neu gap pheps tinh thin lay 2 phan tu

//dau trong stack(chinh la 2 chu ngay trc no) de tao bieu thuc

//xong push nguoc bt vao stack

int main(){

int t;

cin>>t;

while(t--){

string s;cin>>s;

stack<string> a;

for(int i=s.size()-1;i>=0;i--){

if(s[i]=='+'||s[i]=='-'||s[i]=='\*'||s[i]=='/'||s[i]=='^'||s[i]=='%'){

string b=a.top();a.pop();

string c=a.top();a.pop();

string res="(" + b + s[i] + c + ")";

a.push(res);

}

else a.push(string(1,s[i]));

}

cout<<a.top()<<endl;

}

return 0;

}

7.  BIẾN ĐỐI TIỀN TỐ - HẬU TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07010/top)

   Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng hậu tố.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  \*+AB-CD  \*-A/BC-/AKL | AB+CD-\*  ABC/-AK/L-\* |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

string s;cin>>s;

stack<string> a;

for(int i=s.size()-1;i>=0;i--){

if(s[i]=='+'||s[i]=='-'||s[i]=='\*'||s[i]=='/'||s[i]=='^'||s[i]=='%'){

string b=a.top();a.pop();

string c=a.top();a.pop();

string res=b+c+s[i];

a.push(res);

}

else a.push(string(1,s[i]));

}

cout<<a.top()<<endl;

}

return 0;

}

8. TÍNH GIÁ TRỊ BIỂU THỨC HẬU TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07013/top)

Hãy viết chương trình chuyển tính toán giá trị của biểu thức hậu tố.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức hậu tố exp. Các số xuất hiện trong biểu thức là các số đơn có 1 chữ số.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng, chỉ lấy giá trị phần nguyên.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  231\*+9–  875\*+9- | -4  34 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

string s;cin>>s;

stack<long long> a;

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

if(s[i]=='+'||s[i]=='-'||s[i]=='\*'||s[i]=='/'||s[i]=='^'||s[i]=='%'){

long long b=a.top();a.pop();

long long c=a.top();a.pop();

long long res;

if(s[i]=='+') res=c+b;

else if(s[i]=='-') res=c-b;

else if(s[i]=='\*') res=c\*b;

else if(s[i]=='/') res=c/b;

else if(s[i]=='%') res=c%b;

else if(s[i]=='^'){

res=pow(c,b-1);

}

a.push(res);

}

else a.push((long long)(s[i]-'0'));

}

cout<<a.top()<<endl;

}

return 0;

}

9. TÍNH GIÁ TRỊ BIỂU THỨC TIỀN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07014/top)

Hãy viết chương trình tính toán giá trị của biểu thức tiền tố.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp. Các số xuất hiện trong biểu thức là các số đơn có 1 chữ số.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng, chỉ lấy giá trị phần nguyên.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  -+8/632  -+7\*45+20 | 8  25 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

string s;cin>>s;

stack<long long> a;

for(int i=s.size()-1;i>=0;i--){

if(s[i]=='+'||s[i]=='-'||s[i]=='\*'||s[i]=='/'||s[i]=='^'||s[i]=='%'){

long long b=a.top();a.pop();

long long c=a.top();a.pop();

long long res;

if(s[i]=='+') res=b+c;

else if(s[i]=='-') res=b-c;

else if(s[i]=='\*') res=c\*b;

else if(s[i]=='/') res=b/c;

else if(s[i]=='%') res=b%c;

else if(s[i]=='^'){

res=pow(b,c-1);

}

a.push(res);

}

else a.push((long long)(s[i]-'0'));

}

cout<<a.top()<<endl;

}

return 0;

}

10.0. PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07027/top)

Cho dãy số A[] gồm N phần tử. Với mỗi A[i], bạn cần tìm phần tử bên phải đầu tiên lớn hơn nó. Nếu không tồn tại, in ra -1.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (1 ≤ N ≤ 100000).

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

**Output:**

Với mỗi test, in ra trên một dòng N số R[i], với R[i] là giá trị phần tử đầu tiên lớn hơn A[i].

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  4  4 5 2 25  3  2 2 2  4  4 4 5 5 | 5 25 25 -1  -1 -1 -1  5 5 -1 -1 |

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

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

int ok=0;

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

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

cout<<a[j]<<" ";ok=1;

break;

}

}

if(ok==0) cout<<"-1 ";

}

cout<<"-1"<<endl;

}

return 0;

}

11.0.  ĐẢO TỪ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07023/top)

Cho xâu ký tự S. Nhiệm vụ của bạn là đảo ngược các từ trong S. Ví dụ S =  “I like this program very much”, ta nhận được kết quả là “much very program this like I”.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
* T, S thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ Length(S)≤103.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 2 I like this program very much  much very program this like I | much very program this like I I like this program very much |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;cin>>t;

cin.ignore();

while(t--){

string s;

getline(cin,s);

stringstream ss(s);

string w;

vector<string> se;

while(ss>>w){

se.push\_back(w);

}

for(int i=se.size()-1;i>=0;i--){

cout<<se[i]<<" ";

}

cout<<endl;

}

return 0;

}

12.0. THỨ TỰ CẶP DẤU NGOẶC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07040/top)

Cho biểu thức số học có chứa các ký tự dấu ngoặc ‘(‘, ‘)’. Hãy in ra số thứ tự của các cặp ‘(‘, ‘)’ trong biểu thức.

Ví dụ với biểu thức                                              **(a + (b \*c) ) + (d/e)**

ta có thứ tự của các cặp ‘(‘, ‘)’ là                         **1  2  2  1  3  3**

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T (không quá 100).
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một biểu thức số học được đưa vào trên một dòng, độ dài không quá 106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  (a + (b \*c) ) + (d/e)  ( ( () ) ( () ) ) | 1  2  2  1  3  3  1 2  3  3  2  4  5  5  4  1 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main() {

int t; cin >> t;

cin.ignore();

while (t--) {

string s;

getline(cin, s);

stack<int> a;

int x = 1;

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

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

cout << x << " ";

a.push(x);

x++;

}

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

cout << a.top() << " ";

a.pop();

}

}

cout<<endl;

}

return 0;

}

10. HÌNH CHỮ NHẬT LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07019/top)

Cho N cột, mỗi cột có chiều cao bằng H[i]. Bạn hãy tìm hình chữ nhật lớn nhất bị che phủ bởi các cột?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

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

Dòng tiếp theo gồm N số nguyên H[i] (1 ≤ H[i] ≤ 109).

**Output:**

Với mỗi test, in ra diện tích hình chữ nhật lớn nhất tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  7  6 2 5 4 5 1 6  3  2 2 2 | 12  6 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

int n;

cin>>n;

long long a[n],max=0;

for(int i=0;i<n;i++) cin>>a[i];

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

long long l=i-1,r=i+1,dem=1;

while(l>=0&&a[l]>=a[i]){

dem++;l--;

}

while(r<=n-1&&a[r]>=a[i]){

dem++;r++;

}

long long tich=dem\*a[i];

if(tich>max) max=tich;

}

cout<<max<<endl;

}

return 0;

}

19. GIẢI MÃ XÂU KÝ TỰ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07029/top)

Cho xâu ký tự mã hóa str. Hãy viết chương trình giải mã xâu ký tự str. Xâu ký tự mã hóa được thực hiện theo số lần lặp các xâu con của str như sau:

Xâu đầu vào: “abbbababbbababbbab ”

Xâu mã hóa : "3[a3[b]1[ab]]"

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một xâu mã hóa str được viết trên một dòng.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, str thỏa mãn ràng buộc: 1≤T≤100; 1≤length(str)≤100.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 1[b] 3[b2[ca]] | b bcacabcacabcaca |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

string s;cin>>s;

stack<char> res;

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

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

string ss="",top;

int dem=0,tmp=1;

while(res.top()!='['){

top=res.top();res.pop();

ss=top+ss;

}

// loai bo ]

res.pop();

while(!res.empty()&&res.top()>='0'&&res.top()<='9'){

dem+=(res.top()-'0')\*tmp;tmp\*=10;

res.pop();

}

if(dem==0) dem=1;

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

for(auto x : ss) res.push(x);

}

}

else res.push(s[i]);

}

string ans="";

while(!res.empty()){

char x1=res.top();

ans.push\_back(x1);

res.pop();

}

reverse(ans.begin(),ans.end());

cout<<ans<<endl;

}

return 0;

}

Dạng 8: Hàng Đợi

1. CẤU TRÚC DỮ LIỆU HÀNG ĐỢI 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08001/top)

Ban đầu cho một queue rỗng. Bạn cần thực hiện các truy vấn sau:

1. Trả về kích thước của queue
2. Kiểm tra xem queue có rỗng không, nếu có in ra “YES”, nếu không in ra “NO”.
3. Cho một số nguyên và đẩy số nguyên này vào cuối queue.
4. Loại bỏ phần tử ở đầu queue nếu queue không rỗng, nếu rỗng không cần thực hiện.
5. Trả về phần tử ở đầu queue, nếu queue rỗng in ra -1.
6. Trả về phần tử ở cuối queue, nếu queue rỗng in ra -1.

**Dữ liệu vào**

Dòng đầu tiên chứa số nguyên T là số bộ dữ liệu, mỗi bộ dữ theo dạng sau.

Dòng đầu tiên chứa số nguyên n - lượng truy vấn (1 ≤ n ≤ 1000)

N dòng tiếp theo, mỗi dòng sẽ ghi loại truy vấn như trên, với truy vấn loại 3 sẽ có thêm một số nguyên, không quá 106.

**Kết quả:** In ra kết quả của các truy vấn..

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  14  3 1  3 2  3 3  5  6  4  4  4  4  4  3 5  3 6  5  1 | 1  3  5  2 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int a,b=0;cin>>a;

queue<int> res;

while(a--){

int x;

cin>>x;

if(x==1) cout<<res.size()<<endl;

else if(x==2)(res.size()>0)?

//toan tu 3 ngoi kiem tra queue co rong ko

cout<<"NO\n":cout<<"YES\n";

else if(x==3){

int n;cin>>n;

res.push(n);

}

else if(x==4&&res.size()>0) res.pop();

else if(x==5)(res.size()>0)? cout<<res.front()<<endl:cout<<-1<<endl;

else if(x==6)(res.size()>0)? cout<<res.back()<<endl:cout<<-1<<endl;

b=x;

}

if(b==3||b==4) cout<<endl;

}

return 0;

}

2. CẤU TRÚC DỮ LIỆU HÀNG ĐỢI 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08002/top)

Yêu cầu bạn xây dựng một queue với các truy vấn sau đây:

“PUSH x”: Thêm phần tử x vào cuối của queue (0 ≤ x ≤ 1000).

“PRINTFRONT”: In ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

“POP”: Xóa phần tử ở đầu của queue. Nếu queue rỗng, không làm gì cả.

**Dữ liệu vào:**

Dòng đầu tiên là số lượng truy vấn Q (Q ≤ 100000).

Mỗi truy vấn có dạng như trên.

**Kết quả:**

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9  PUSH 1  PUSH 2  POP  PRINTFRONT  PUSH 3  PRINTFRONT  POP  POP  PRINTFRONT | 2  2  NONE |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

queue<int> res;

while(t--){

string x;cin>>x;

if(x=="PUSH"){

int n;cin>>n;

res.push(n);

cout<<endl;

}

else if(x=="PRINTFRONT")(res.size()>0)? cout<<res.front()<<endl:cout<<"NONE"<<endl;

else if(x=="POP"){

if(res.size()>0) res.pop();

cout<<endl;

}

}

return 0;

}

3.  HÀNG ĐỢI HAI ĐẦU (DEQUEUE)

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08003/top)

Yêu cầu bạn xây dựng một hàng đợi hai đầu với các truy vấn sau đây:

“PUSHFRONT x”: Thêm phần tử x vào đầu của dequeue (0 ≤ x ≤ 1000).

“PRINTFRONT”: In ra phần tử đầu tiên của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPFRONT”: Xóa phần tử đầu của dequeue. Nếu dequeue rỗng, không làm gì cả.

“PUSHBACK x”: Thêm phần tử x vào cuối của dequeue (0 ≤ x ≤ 1000).

“PRINTBACK”: In ra phần tử cuối của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPBACK”: Xóa phần tử cuối của dequeue. Nếu dequeue rỗng, không làm gì cả.

**Dữ liệu vào:**

Dòng đầu tiên là số lượng truy vấn Q (Q ≤ 100000).

Mỗi truy vấn có dạng như trên.

**Kết quả:**

Với mỗi truy vấn “PRINTFRONT” và “PRINTBACK”, hãy in ra kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  PUSHBACK 1  PUSHFRONT 2  PUSHBACK 3  PRINTFRONT  POPFRONT  PRINTFRONT  POPFRONT  PRINTBACK  POPFRONT  PRINTBACK | 2  1  3  NONE |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

deque<int> res;

while(t--){

string x;cin>>x;

if(x=="PUSHFRONT"){

int n;cin>>n;res.push\_front(n);cout<<endl;

}

else if(x=="PRINTFRONT")(res.size()>0)? cout<<res.front()<<endl:cout<<"NONE"<<endl;

else if(x=="POPFRONT"&&res.size()>0) res.pop\_front();

else if(x=="PUSHBACK"){

int n;cin>>n;res.push\_back(n);

}

else if(x=="PRINTBACK")(res.size()>0)? cout<<res.back()<<endl:cout<<"NONE"<<endl;

else if(x=="POPBACK"&&res.size()>0) res.pop\_back();

}

return 0;

}

4. SỐ NHỊ PHÂN TỪ 1 ĐẾN N

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08005/top)

Cho số tự nhiên n. Hãy in ra tất cả các số nhị phân từ 1 đến n.

**Input:**

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Mỗi test là một số tự nhiên n được ghi trên một dòng (n≤10000).

**Output:**

* Đưa ra kết quả mỗi test trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  5 | 1 10  1 10 11 100 101 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n,a[100],m;

void cd(int i){

m=0;

while(i!=0){

a[m++]=i%2;

i/=2;

}

for(int j=m-1;j>=0;j--) cout<<a[j];

cout<<" ";

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

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

cd(i);

}

cout<<endl;

}

return 0;

}

5. SỐ 0 VÀ SỐ 9

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08006/top)

Cho số tự nhiên N. Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 9 và số 0 chia hết cho N. Ví dụ với N = 5 ta sẽ tìm ra  X = 90.

**Input:**

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Những dòng kế tiếp mỗi dòng ghi lại một test. Mỗi test là một số tự nhiên N được ghi trên một dòng (N≤100).

**Output:**

* Đưa ra theo từng dòng số X nhỏ nhất chia hết cho N tìm được .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5  7 | 90  9009 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

string in(){

queue<string> res;

res.push("9");

while(res.size()>0){

string ans=res.front();res.pop();

long long tmp=0;

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

tmp=tmp\*10+(int)(ans[i]-'0');

tmp%=n;

}

if(tmp==0) return ans;

res.push(ans+"0");

res.push(ans+"9");

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

cout<<in()<<endl;

}

return 0;

}

6. SỐ BDN 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08007/top)

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số K = 1, 10, 101. Cho số tự nhiên N (N<263). Hãy cho biết có bao nhiêu số BDN nhỏ hơn N. Ví dụ N=100 ta có 4 số BDN bao gồm các số: 1, 10, 11, 100.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
* T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  10  100  200 | 2  4  7 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

long long n;cin>>n;

vector<long long> a;

while(n){

a.push\_back(n%10);

n/=10;

}

//dao nguoc vector

reverse(a.begin(),a.end());

int x=a.size();

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

if(a[i]>1){

x=i;break;

}

}

for(int i=x;i<a.size();i++){

a[i]=1;

}

//chuyen doi sang he thap phan

long long res=0,y=1;

for(int i=a.size()-1;i>=0;i--){

res+=a[i]\*y;

y\*=2;

}

cout<<res<<endl;

}

return 0;

}

7. SỐ BDN 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08008/top)

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số K =  101 là số BDN, k=102 không phải là số BDN.

Số BDN của N là số P =M´N sao cho P là số BDN. Cho số tự nhiên N (N<1000), hãy tìm số BDN nhỏ nhất của N.

**Ví dụ**. Với N=2, ta tìm được số BDN của N là P = 5´2=10. N = 17 ta tìm được số BDN của 17 là P = 653´17=11101.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
* T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N.

**Output:**

                        Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2  12  17 | 10  11100  11101 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

string in(){

queue<string> res;

res.push("1");

while(res.size()>0){

string ans=res.front();res.pop();

long long tmp=0;

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

tmp=tmp\*10+(int)(ans[i]-'0');

tmp%=n;

}

if(tmp==0) return ans;

res.push(ans+"0");

res.push(ans+"1");

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

cout<<in()<<endl;

}

return 0;

}

8. SỐ LỘC PHÁT 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08017/top)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test (T<10);
* T dòng kế tiếp mỗi dòng ghi số N (1<N<15).

**Output:**

* In ra đáp án theo thứ tự giảm dần.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 88 86 68 66 8 6  888 886 868 866 688 686 668 666 88 86 68 66 8 6 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

vector<string> res;

int n,X[100],m;

bool ok;

void sinh(int i){

while((i>0)&&(X[i]==1)) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n; j++) X[j]=1-X[j];

return;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

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

ok=false;

for(int j=1;j<=i;j++) X[j]=0;

while(!ok){

string s;

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

if(X[j]==1) s+='8';

else s+='6';

}

res.push\_back(s);

sinh(i);

}

for(int i=res.size()-1;i>=0;i--){

cout<<res[i]<<" ";

}

res.clear();

}

cout<<endl;

}

return 0;

}

9. SỐ LỘC PHÁT 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08018/top)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test (T<10);
* T dòng kế tiếp mỗi dòng ghi số N (1<N<15).

**Output:**

* Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in đáp án **theo thứ tự tăng dần.**

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 6  6 8 66 68 86 88  14  6 8 66 68 86 88 666 668 686 688 866 868 886 888 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

vector<string> res;

int n,X[1000],m;

bool ok;

void sinh(int i){

while((i>0)&&(X[i]==1)) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n; j++) X[j]=1-X[j];

return;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

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

ok=false;

for(int j=1;j<=i;j++) X[j]=0;

while(!ok){

string s;

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

if(X[j]==1) s+='8';

else s+='6';

}

res.push\_back(s);

sinh(i);

}

}

cout<<res.size()<<endl;

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

cout<<res[i]<<" ";

}

res.clear();

cout<<endl;

}

return 0;

}

10. SỐ LỘC PHÁT 3

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08019/top)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test (T<10);
* T dòng kế tiếp mỗi dòng ghi số N (1<N<15).

**Output:**

* Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in ra đáp án **theo thứ tự giảm dần**.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 6  88 86 68 66 8 6  14  888 886 868 866 688 686 668 666 88 86 68 66 8 6 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

vector<string> res;

vector<string> ans;

int n,X[100],m;

bool ok;

void sinh(int i){

while((i>0)&&(X[i]==1)) i--;

if(i==0){

ok=true;return;

}

else for(int j=i; j<=n; j++) X[j]=1-X[j];

return;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n;

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

ok=false;

for(int j=1;j<=i;j++) X[j]=0;

while(!ok){

string s;

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

if(X[j]==1) s+='8';

else s+='6';

}

res.push\_back(s);

sinh(i);

}

for(int i=res.size()-1;i>=0;i--){

ans.push\_back(res[i]);

}

res.clear();

}

cout<<ans.size()<<endl;

for(int i=0;i<ans.size();i++) cout<<ans[i]<<" ";

ans.clear();

cout<<endl;

}

return 0;

}

11. GIÁ TRỊ NHỎ NHẤT CỦA XÂU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08004/top)

Cho xâu ký tự S[] bao gồm các ký tự in hoa [A, B, …,Z]. Ta định nghĩa giá trị của xâu S[] là tổng bình phương số lần xuất hiện mỗi ký tự trong xâu. Ví dụ với xâu S[] = “AAABBCD” ta có F(S) = 32 + 22 + 12 + 12 = 15. Hãy tìm giá trị nhỏ nhất của xâu S[] sau khi loại bỏ K ký tự trong xâu.

**Input**:

* Dòng đầu tiên đưa vào số lượng test T (T≤100).
* Mỗi test được tổ chức thành 2 dòng. Dòng thứ nhất ghi lại số K. Dòng thứ 2 ghi lại xâu ký tự S[] có độ dài không vượt quá 10^6.

Output:

* Đưa ra giá trị nhỏ nhất của mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2  0  ABCC  1  ABCC | 6  3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int k;

int main(){

int t;

cin >> t;

while(t--){

cin>>k;

string s;cin>>s;

//xay dung hang doi uu tien

long long dem[300]={0};

priority\_queue<long long,vector<long long>> res;

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

dem[s[i]]++;

}

for(int i=0;i<260;i++){

if(dem[i]>0) res.push(dem[i]);

}

while(res.size()>0&&k>0){

long long x=res.top();res.pop();

k--;

x--;

res.push(x);

}

long long ans=0;

while(res.size()>0){

ans+=res.top()\*res.top();

res.pop();

}

cout<<ans<<endl;

}

return 0;

}

12. BIẾN ĐỔI S – T

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA08009/top)

Cho hai số nguyên dương S và T (S, T<10000) và hai thao tác (a), (b) dưới đây:

**Thao tác (a)**: Trừ S đi 1  (S = S-1) ;

**Thao tác (b)**: Nhân S với 2 ( S = S\*2);

Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất. Ví dụ với    S =2, T=5 thì số các bước ít nhất để dịch chuyển S thành T thông qua 4 thao tác sau:

**Thao tác (a)**: 2\*2 = 4;

**Thao tác (b)**: 4-1 = 3;

**Thao tác (a)**: 3\*2 = 6;

**Thao tác (b)**: 6-1 = 5;

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
* T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một bộ đôi S và T.

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2 5  3 7  7 4 | 4  4  3 |

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int s,t;

struct G{

int x,dem;

};

int BFS(int a,int b){

set<int> res;

res.insert(a);

queue<G> q;

q.push({a,0});

while(q.size()>0){

G ans=q.front();q.pop();

if(ans.x==b) return ans.dem;

if(ans.x\*2==b||ans.x==b+1) return ans.dem+1;

if(res.find(ans.x\*2)==res.end()&&ans.x<b){

q.push({ans.x\*2,ans.dem+1});

res.insert(ans.x\*2);

}

if(res.find(ans.x-1)==res.end()&&ans.x-1>0){

q.push({ans.x-1,ans.dem+1});

res.insert(ans.x-1);

}

}

}

int main(){

int T;

cin >> T;

while(T--){

cin>>s>>t;

cout<<BFS(s,t)<<endl;

}

return 0;

}

Dạng 9: DUYỆT ĐỒ THỊ

1. CHUYỂN DANH SÁCH CẠNH SANG DANH SÁCH KỀ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09001/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết chương trình thực hiện chuyển đổi biểu diễn đồ thị dưới dạng danh sách kề.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| +1 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh của đồ thị; |E| dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách kề của các đỉnh tương ứng theo khuôn dạng của ví dụ dưới đây. Các đỉnh trong danh sách in ra theo thứ tự tăng dần.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int V,E;cin>>V>>E;

// xay dung vecor 2 chieu luu canh ke cua moi dinh

vector<int> ke[V+1];

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

int x,y;cin>>x>>y;

ke[x].push\_back(y);

ke[y].push\_back(x);

}

for(int i=1;i<=V;i++) sort(ke[i].begin(),ke[i].end());

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

cout<<i<<": ";

for(int j=0;j<ke[i].size();j++) cout<<ke[i][j]<<" ";

cout<<endl;

}

}

return 0;

}

2. CHUYỂN TỪ DANH SÁCH KỀ SANG DANH SÁCH CẠNH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09002/top)

Cho đơn đồ thị G vô hướng liên thông được mô tả bởi danh sách kề. Hãy in ra danh sách cạnh tương ứng của G.

**Input**

* Dòng đầu tiên ghi số N là số đỉnh (1<N<50)
* N dòng tiếp theo mỗi dòng ghi 1 danh sách kề lần lượt theo thứ tự từ đỉnh 1 đến đỉnh N

**Output:**Ghi ra lần lượt từng cạnh của đồ thị theo thứ tự tăng dần.

**Ví dụ**

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

// by hung nguyen

#include <iostream>

#include <string>

#include <vector>

#include <cstring>

using namespace std;

int main() {

3. BIỂU DIỄN ĐỒ THỊ CÓ HƯỚNG.

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09003/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết chương trình thực hiện chuyển đổi biểu diễn đồ thị dưới dạng danh sách kề.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| +1 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh của đồ thị; |E| dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách kề của các đỉnh tương ứng theo khuôn dạng của ví dụ dưới đây. Các đỉnh trong danh sách in ra theo thứ tự tăng dần.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 1  6  9  1  2  2  5  3  1  3  2  3  5  4  3  5  4  5  6  6  4 | 1: 2  2: 5  3: 1 2 5  4: 3  5: 4 6  6: 4 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

while(t--){

int V,E;cin>>V>>E;

// xay dung vecor 2 chieu luu canh ke cua moi dinh

vector<int> ke[V+1];

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

int x,y;cin>>x>>y;

ke[x].push\_back(y);

}

for(int i=1;i<=V;i++) sort(ke[i].begin(),ke[i].end());

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

cout<<i<<": ";

for(int j=0;j<ke[i].size();j++) cout<<ke[i][j]<<" ";

cout<<endl;

}

}

return 0;

}

4.  DFS TRÊN ĐỒ THỊ VÔ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09004/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết thuật toán duyệt theo chiều sâu bắt đầu tại đỉnh uV (DFS(u)=?)

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| +1 dòng: dòng đầu tiên đưa vào ba số |V|, |E| tương ứng với số đỉnh và số cạnh của đồ thị, và u là đỉnh xuất phát; |E| dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh được duyệt theo thuật toán DFS(u) của mỗi test theo khuôn dạng của ví dụ dưới đây.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 1  6 9 5  1 2  1 3  2 3  2 4  3 4  3 5  4 5  4 6  5 6 | 5 3 1 2 4 6 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dau;

vector<int> ke[1001];

bool chuaxet[1001];

void init(){

for(int i=0;i<=1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void DFS(int u){

chuaxet[u]=false;

cout<<u<<" ";

for(int i=0;i<ke[u].size();i++){

if(chuaxet[ke[u][i]]) DFS(ke[u][i]);

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>a>>b>>dau;

init();

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

int u,v;

cin>>u>>v;

ke[u].push\_back(v);

ke[v].push\_back(u);

}

DFS(dau);

cout<<endl;

}

return 0;

}

5.  LIỆT KÊ CẠNH CẦU VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09013/top)

Cho đồ thị vô hướng liên thông G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán BFS, hãy đưa ra tất cả các cạnh cầu của đồ thị?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| + 1 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; |E| dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các cạch cầu của mỗi test theo từng dòng. In ra đáp án theo thứ tự từ điển, theo dạng “a b …” với a < b.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dem;

bool chuaxet[1005];

vector<int> ke[1005];

vector<pair<int,int>> ans;

void reset(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

int TPLT(){

int soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

return soTPLT;

}

void canhcau(int canh,int tmp,vector<pair<int,int>> res){

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

if(i!=canh){

ke[res[i].first].push\_back(res[i].second);

ke[res[i].second].push\_back(res[i].first);

}

}

int dem=TPLT();

if(dem>tmp){

if(res[canh].first<res[canh].second)

ans.push\_back({res[canh].first,res[canh].second});

else ans.push\_back({res[canh].second,res[canh].first});

}

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

reset();ans.clear();

vector<pair<int,int>>res;

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

int u,v;

cin>>u>>v;

res.push\_back({u,v});

ke[v].push\_back(u);

ke[u].push\_back(v);

}

int tmp=TPLT();

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

reset();

canhcau(i,tmp,res);

}

sort(ans.begin(),ans.end());

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

cout<<ans[i].first<<" "<<ans[i].second<<" ";

cout<<endl;

cout<<endl;

}

return 0;

}

6. ĐƯỜNG ĐI VÀ CHU TRÌNH EULER VỚI ĐỒ THỊ VÔ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA10004/top)

Cho đồ thị vô hướng liên thông G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy kiểm tra xem đồ thị có đường đi Euler hay chu trình Euler hay không?

Đường đi Euler bắt đầu tại một đỉnh, và kết thúc tại một đỉnh khác.

Chu trình Euler bắt đầu tại một đỉnh, và kết thúc chu trình tại chính đỉnh đó.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra 1, 2, 0 kết quả mỗi test theo từng dòng tương ứng với đồ thị có đường đi Euler, chu trình Euler và trường hợp không tồn tại.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

int TPLT(){

int soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

return soTPLT;

}

void check(){

int dem=0;

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

int x=ke[i].size();

if(x%2!=0) dem++;

}

if(dem==0) cout<<"2";

else if(dem==2) cout<<"1";

else cout<<"0";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

if(TPLT()!=1) cout<<0;

else{

check();

}

cout<<endl;

}

return 0;

}

7.  CHU TRÌNH EULER TRONG ĐỒ THỊ CÓ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA10005/top)

Cho đồ thị có hướng liên thông yếu G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy kiểm tra xem đồ thị có chu trình Euler hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra 1, 0 kết quả mỗi test theo từng dòng tương ứng với đồ thị có chu trình Euler và trường hợp không tồn tại đáp án.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 2  6  10  1 2 2 4 2 5 3 1 3 2 4 3 4 5 5 3 5 6 6 4  3 3  1 2 2 3 1 3 | 1  0 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dem[1005];

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

dem[i]={0};

}

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

int TPLT(){

int soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

return soTPLT;

}

void check(){

int tmp1=0,tmp2=0,tmp3=0;

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

int x=ke[i].size();

if(dem[i]%x!=0||dem[i]!=x\*2){

cout<<"0";return;

}

}

cout<<"1";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

dem[u]++;

dem[v]++;

ke[u].push\_back(v);

//ke[u].push\_back(v);

}

if(TPLT()!=1) cout<<0;

else{

check();

}

cout<<endl;

}

return 0;

}

8. CHUYỂN MA TRẬN KỀ SANG DANH SÁCH KỀ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09021/top)

Ma trận kề A của một đồ thị vô hướng là một ma trận chỉ có các số 0 hoặc 1 trong đó A[i][j] = 1 có ý nghĩa là đỉnh i kề với đỉnh j (chỉ số tính từ 1).

Danh sách kề thì liệt kê các đỉnh kề với đỉnh đó theo thứ tự tăng dần.

Hãy chuyển biểu diễn đồ thị từ dạng ma trận kề sang dạng danh sách kề.

**Input:** Dòng đầu tiên chứa số nguyên n – số đỉnh của đồ thị (1 < n ≤ 1000). n dòng tiếp theo, mỗi dòng có n số nguyên có giá trị 0 và 1 mô tả ma trận kề của đồ thị.

**Output:** Gồm n dòng, dòng thứ i chứa các số nguyên là đỉnh có nối với đỉnh i và được sắp xếp tăng dần. Dữ liệu đảm bảo mỗi đỉnh có kết nối với ít nhất 1 đỉnh khác.

**Ví dụ:**

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

// by hung nguyen 2

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;cin>>n;

vector<int> ke[n+1];

int a[n][n];

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

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

cin>>a[i][j];

if(a[i][j]==1) ke[i].push\_back(j+1);

}

}

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

for(int j=0;j<ke[i].size();j++) cout<<ke[i][j]<<" ";

cout<<endl;

}

return 0;

}

9.  CHUYỂN DANH SÁCH KỀ SANG MA TRẬN KỀ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09020/top)

Cho đơn đồ thị vô hướng có n đỉnh dưới dạng danh sách kề.

Hãy biểu diễn đồ thị bằng ma trận kề.

**Input:** Dòng đầu tiên chứa số nguyên n – số đỉnh của đồ thị (1 ≤  n ≤  1000). n dòng tiếp theo, dòng thứ i chứa các số nguyên là các đỉnh kề với đỉnh i.

**Output:** Ma trận kề của đồ thị.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a[1000][1000];

int main() {

int n; cin >> n;

string str;

cin.ignore();

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

getline(cin, str);

str += " ";

int res = 0;

int j = 0;

while (j < str.size()) {

if (str[j] >= '0' && str[j] <= '9') {

res= res \* 10 + (int)(str[j] - '0');

}

else if (res > 0) {

a[i][res-1]=1;

res=0;

}

j++;

}

}

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

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

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

}

cout<<endl;

}

return 0;

}

10. DFS TRÊN ĐỒ THỊ CÓ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09022/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết thuật toán duyệt theo chiều sâu bắt đầu tại đỉnh uV (DFS(u)=?)

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào ba số |V|, |E|, uV tương ứng với số đỉnh, số cạnh và đỉnh bắt đầu duyệt; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh được duyệt theo thuật toán DFS(u) của mỗi test theo khuôn dạng của ví dụ dưới đây.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

bool chuaxet[1001];

vector<int> ke[1000];

void init(){

for(int i=0;i<=1000;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void DFS(int u){

chuaxet[u]=false;

cout<<u<<" ";

for(int i=0;i<ke[u].size();i++){

if(chuaxet[ke[u][i]]) DFS(ke[u][i]);

}

}

int main(){

int t;

cin >> t;

while(t--){

int V,E,u;

cin>>V>>E>>u;

init();

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

int a,b;

cin>>a>>b;

ke[a].push\_back(b);

}

DFS(u);

cout<<endl;

}

return 0;

}

11.  BFS TRÊN ĐỒ THỊ CÓ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09024/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết thuật toán duyệt theo chiều rộng bắt đầu tại đỉnh uV (BFS(u)=?)

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào ba số |V|, |E|, uV tương ứng với số đỉnh, số cạnh và đỉnh bắt đầu duyệt; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh được duyệt theo thuật toán BFS(u) của mỗi test theo khuôn dạng của ví dụ dưới đây.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dau;

vector<int> ke[1001];

bool chuaxet[1001];

void init(){

for(int i=0;i<=1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> res;

res.push(u);

while(!res.empty()){

int s=res.front();res.pop();

chuaxet[s]=false;

cout<<s<<" ";

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

if(chuaxet[ke[s][i]]){

chuaxet[ke[s][i]]=false;

res.push(ke[s][i]);

}

}

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>a>>b>>dau;

init();

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

int u,v;

cin>>u>>v;

ke[u].push\_back(v);

// ke[v].push\_back(u);

}

BFS(dau);

cout<<endl;

}

return 0;

}

12. TÌM ĐƯỜNG ĐI THEO DFS VỚI ĐỒ THỊ CÓ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09025/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi từ đỉnh sV đến đỉnh tV trên đồ thị bằng thuật toán DFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào bốn số |V|, |E|, sV, tV tương ứng với số đỉnh, số cạnh,  đỉnh u, đỉnh v; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra đường đi từ đỉnh s đến đỉnh t của mỗi test theo thuật toán DFS của mỗi test theo khuôn dạng của ví dụ dưới đây. Nếu không có đáp án, in ra -1.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,s,t,truoc[1005];

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

truoc[i]=0;

}

}

void DFS(int u,int v){

if(chuaxet[v]==false) return;

chuaxet[u]=false;

for(int i=0;i<ke[u].size();i++){

if(chuaxet[ke[u][i]]){

truoc[ke[u][i]]=u;

DFS(ke[u][i],v);

}

}

}

void check(int u,int v){

if(chuaxet[v]){

cout<<-1;return;

}

vector<int> res;

while(u!=v){

if(u==0){

cout<<"-1";return;

}

res.push\_back(u);

u=truoc[u];

}

res.push\_back(v);

reverse(res.begin(),res.end());

for(int i=0;i<res.size();i++) cout<<res[i]<<" ";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b>>s>>t;

init();

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

int u,v;

cin>>u>>v;

ke[u].push\_back(v);

//ke[u].push\_back(v);

}

DFS(s,t);

check(t,s);

cout<<endl;

}

return 0;

}

13.  ĐƯỜNG THI THEO BFS TRÊN ĐỒ THỊ CÓ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09026/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi từ đỉnh uV đến đỉnh vV trên đồ thị bằng thuật toán BFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào bốn số |V|, |E|, sV, tV tương ứng với số đỉnh, số cạnh,  đỉnh u, đỉnh v; |E| Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra đường đi từ đỉnh s đến đỉnh t của mỗi test theo thuật toán BFS của mỗi test theo khuôn dạng của ví dụ dưới đây. Nếu không có đáp án, in ra -1.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,s,t,truoc[1005];

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

truoc[i]=0;

}

}

void BFS(int u,int v){

if(!chuaxet[v]) return;

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

truoc[ke[top][i]]=top;

k.push(ke[top][i]);

}

}

}

}

void check(int u,int v){

if(chuaxet[v]){

cout<<-1;return;

}

vector<int> res;

while(u!=v){

if(u==0){

cout<<"-1";return;

}

res.push\_back(u);

u=truoc[u];

}

res.push\_back(v);

reverse(res.begin(),res.end());

for(int i=0;i<res.size();i++) cout<<res[i]<<" ";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b>>s>>t;

init();

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

int u,v;

cin>>u>>v;

//ke[v].push\_back(u);

ke[u].push\_back(v);

}

BFS(s,t);

check(t,s);

cout<<endl;

}

return 0;

}

14. KIỂM TRA ĐƯỜNG ĐI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09027/top)

Cho đồ thị vô hướng có N đỉnh và M cạnh. Có Q truy vấn, mỗi truy vấn yêu cầu trả lời câu hỏi giữa 2 đỉnh x và y có tồn tại đường đi tới nhau hay không?

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test gồm 2 số nguyên N, M (1 ≤ N, M ≤ 1000).
* M dòng tiếp theo, mỗi dòng gồm 2 số nguyên u, v cho biết có cạnh nối giữa đỉnh u và v.
* Dòng tiếp là số lượng truy vấn Q (1 ≤ Q ≤ 1000).
* Q dòng tiếp theo, mỗi dòng gồm 2 số nguyên x và y.

**Output:**Với mỗi truy vấn, in ra “YES” nếu có đường đi từ x tới y, in ra “NO” nếu ngược lại.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dau;

vector<int> ke[1001];

bool chuaxet[1001];

void init(){

for(int i=0;i<=1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u,int v){

queue<int> res;

res.push(u);

while(!res.empty()){

int s=res.front();res.pop();

chuaxet[s]=false;

if(chuaxet[v]==false){

cout<<"YES"<<endl;return;

}

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

if(chuaxet[ke[s][i]]){

chuaxet[ke[s][i]]=false;

res.push(ke[s][i]);

}

}

}

cout<<"NO"<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ke[u].push\_back(v);

ke[v].push\_back(u);

}

int Q;cin>>Q;

while(Q--){

int x,y;cin>>x>>y;

BFS(x,y);

memset(chuaxet,true,sizeof(chuaxet));

}

}

return 0;

}

15.  BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09005/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết thuật toán duyệt theo chiều rộng bắt đầu tại đỉnh uV (BFS(u)=?)

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào ba số |V|, |E|, uV tương ứng với số đỉnh, số cạnh và đỉnh bắt đầu duyệt; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤200; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh được duyệt theo thuật toán BFS(u) của mỗi test theo khuôn dạng của ví dụ dưới đây.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dau;

vector<int> ke[1001];

bool chuaxet[1001];

void init(){

for(int i=0;i<=1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> res;

res.push(u);

while(!res.empty()){

int s=res.front();res.pop();

chuaxet[s]=false;

cout<<s<<" ";

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

if(chuaxet[ke[s][i]]){

chuaxet[ke[s][i]]=false;

res.push(ke[s][i]);

}

}

}

}

int main(){

int t;

cin >> t;

while(t--){

cin>>a>>b>>dau;

init();

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

int u,v;

cin>>u>>v;

ke[u].push\_back(v);

ke[v].push\_back(u);

}

BFS(dau);

cout<<endl;

}

return 0;

}

16.  ĐƯỜNG ĐI THEO DFS VỚI ĐỒ THỊ VÔ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09006/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi từ đỉnh sV đến đỉnh tV trên đồ thị bằng thuật toán DFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào bốn số |V|, |E|, sV, tV tương ứng với số đỉnh, số cạnh,  đỉnh u, đỉnh v; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra đường đi từ đỉnh s đến đỉnh t của mỗi test theo thuật toán DFS của mỗi test theo khuôn dạng của ví dụ dưới đây. Nếu không có đáp án, in ra -1.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,s,t,truoc[1005];

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

truoc[i]=0;

}

}

void DFS(int u,int v){

if(chuaxet[v]==false) return;

chuaxet[u]=false;

for(int i=0;i<ke[u].size();i++){

if(chuaxet[ke[u][i]]){

truoc[ke[u][i]]=u;

DFS(ke[u][i],v);

}

}

}

void check(int u,int v){

if(chuaxet[v]){

cout<<-1;return;

}

vector<int> res;

while(u!=v){

res.push\_back(u);

u=truoc[u];

}

res.push\_back(v);

reverse(res.begin(),res.end());

for(int i=0;i<res.size();i++) cout<<res[i]<<" ";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b>>s>>t;

init();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

DFS(s,t);

check(t,s);

cout<<endl;

}

return 0;

}

17.  ĐƯỜNG ĐI THEO BFS TRÊN ĐỒ THỊ VÔ HƯỚNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09007/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi từ đỉnh sV đến đỉnh tV trên đồ thị bằng thuật toán BFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào bốn số |V|, |E|, sV, tV tương ứng với số đỉnh, số cạnh,  đỉnh u, đỉnh v; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra đường đi từ đỉnh s đến đỉnh t của mỗi test theo thuật toán BFS của mỗi test theo khuôn dạng của ví dụ dưới đây. Nếu không có đáp án, in ra -1.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,s,t,truoc[1005];

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

truoc[i]=0;

}

}

void BFS(int u,int v){

if(!chuaxet[v]) return;

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

truoc[ke[top][i]]=top;

k.push(ke[top][i]);

}

}

}

}

void check(int u,int v){

if(chuaxet[v]){

cout<<-1;return;

}

vector<int> res;

while(u!=v){

res.push\_back(u);

u=truoc[u];

}

res.push\_back(v);

reverse(res.begin(),res.end());

for(int i=0;i<res.size();i++) cout<<res[i]<<" ";

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b>>s>>t;

init();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

BFS(s,t);

check(t,s);

cout<<endl;

}

return 0;

}

18.  ĐẾM SỐ THÀNH PHẦN LIÊN THÔNG

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09008/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm số thành phần liên thông của đồ thị.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra số thành phần liên thông của đồ thị.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

void TPLT(){

int soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

cout<<soTPLT<<endl;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

TPLT();

}

return 0;

}

19.  TÌM SỐ THÀNH PHẦN LIÊN THÔNG VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09009/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy tìm số thành phần liên thông của đồ thị bằng thuật toán BFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra số thành phần liên thông của đồ thị bằng thuật toán BFS.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

chuaxet[i]=true;

}

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

void TPLT(){

int soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

cout<<soTPLT<<endl;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

TPLT();

}

return 0;

}

20.  KIỂM TRA TÍNH LIÊN THÔNG MẠNH VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09010/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán BFS, hãy kiểm tra xem đồ thị có liên thông mạnh hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra “YES”, hoặc “NO” theo từng dòng tương ứng với test là liên thông mạnh hoặc không liên thông mạnh.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1  6 9  1 2 2 4 3 1 3 2 3 5 4 3 5 4 5 6 6  3 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,dem;

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

}

}

void reset(){

for(int i=0;i<1001;i++) chuaxet[i]=true;

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

dem++;

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

bool ktra(){

reset();

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

dem=0;BFS(i);

if(dem!=a) return false;

else reset();

}

return true;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();reset();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

//ke[u].push\_back(v);

}

if(ktra()) cout<<"YES";

else cout<<"NO";

cout<<endl;

}

return 0;

}

21.  LIỆT KÊ ĐỈNH TRỤ VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09012/top)

Cho đồ thị vô hướng liên thông G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán BFS, hãy đưa ra tất cả các đỉnh trụ của đồ thị?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh trụ của mỗi test  theo từng dòng.

**Ví dụ:**

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

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b,soTPLT,dem;

bool chuaxet[1005];

vector<int> ke[1005];

void init(){

for(int i=0;i<1001;i++){

ke[i].clear();

}

}

void reset(){

for(int i=0;i<1001;i++) chuaxet[i]=true;

}

void BFS(int u){

queue<int> k;

k.push(u);

chuaxet[u]=false;

while(!k.empty()){

int top=k.front();

k.pop();

for(int i=0;i<ke[top].size();i++){

if(chuaxet[ke[top][i]]){

chuaxet[ke[top][i]]=false;

k.push(ke[top][i]);

}

}

}

}

void dinhtru(){

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

reset();

chuaxet[i]=false;

dem=0;

for(int j=1;j<=a;j++){

if(chuaxet[j]){

dem++;

BFS(j);

}

}

if(dem>soTPLT) cout<<i<<" ";

}

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();reset();

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

int u,v;

cin>>u>>v;

ke[v].push\_back(u);

ke[u].push\_back(v);

}

soTPLT=0;

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

if(chuaxet[i]){

soTPLT++;

BFS(i);

}

}

dinhtru();

cout<<endl;

}

return 0;

}

22. KIỂM TRA CHU TRÌNH VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09014/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán BFS, hãy kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output: |
| 1  6  9  1 2 1 3 2 3 2 5 3 4 3 5 4 5 4 6 5 6 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

int ma[1005][1005];

void init(){

memset(chuaxet,true,sizeof(chuaxet));

memset(ma,0,sizeof(ma));

}

int DFS(int u){

stack<int> k;

k.push(u);

int truoc[a+5];

memset(truoc,0,sizeof(truoc));

chuaxet[u]=false;

while(!k.empty()){

int top=k.top();

k.pop();

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

if(ma[top][i]==1&&chuaxet[i]){

chuaxet[i]=false;

k.push(top);

k.push(i);

truoc[i]=top;

break;

}

else if(chuaxet[i]==false&&i!=truoc[top]&&ma[top][i]==1&&i==u){

return 1;

}

}

}

return 0;

}

int check(){

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

memset(chuaxet,true,sizeof(chuaxet));

if(DFS(i)) return 1;

}

return 0;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ma[u][v]=1;

ma[v][u]=1;

}

if(check()) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

return 0;

}

23. KIỂM TRA CHU TRÌNH SỬ DỤNG DISJOIN SET

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA10003/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng **Disjoin** **Set**, hãy kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1  6 9  1 2 1 3 2 3 2 5 3 4 3 5 4 5 4 6 5 6 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

int ma[1005][1005];

void init(){

memset(chuaxet,true,sizeof(chuaxet));

memset(ma,0,sizeof(ma));

}

int DFS(int u){

stack<int> k;

k.push(u);

int truoc[a+5];

memset(truoc,0,sizeof(truoc));

chuaxet[u]=false;

while(!k.empty()){

int top=k.top();

k.pop();

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

if(ma[top][i]==1&&chuaxet[i]){

chuaxet[i]=false;

k.push(top);

k.push(i);

truoc[i]=top;

break;

}

else if(chuaxet[i]==false&&i!=truoc[top]&&ma[top][i]==1&&i==u){

return 1;

}

}

}

return 0;

}

int check(){

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

memset(chuaxet,true,sizeof(chuaxet));

if(DFS(i)) return 1;

}

return 0;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ma[u][v]=1;

ma[v][u]=1;

}

if(check()) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

return 0;

}

24. KIỂM TRA CHU TRÌNH TRÊN ĐỒ THỊ CÓ HƯỚNG VỚI DFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09016/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán DFS, hãy kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1  6 9  1 2 2 4 3 1 3 2 3 5 4 3 5 4 5 6 6 4 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

int ma[1005][1005];

void init(){

memset(chuaxet,true,sizeof(chuaxet));

memset(ma,0,sizeof(ma));

}

int DFS(int u){

stack<int> k;

k.push(u);

int truoc[a+5];

memset(truoc,0,sizeof(truoc));

chuaxet[u]=false;

while(!k.empty()){

int top=k.top();

k.pop();

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

if(ma[top][i]==1&&chuaxet[i]){

chuaxet[i]=false;

k.push(top);

k.push(i);

truoc[i]=top;

break;

}

else if(chuaxet[i]==false&&i!=truoc[top]&&ma[top][i]==1&&i==u){

return 1;

}

}

}

return 0;

}

int check(){

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

memset(chuaxet,true,sizeof(chuaxet));

if(DFS(i)) return 1;

}

return 0;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ma[u][v]=1;

// ma[v][u]=1;

}

if(check()) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

return 0;

}

25. KIỂM TRA CHU TRÌNH TRÊN ĐỒ THỊ CÓ HƯỚNG VỚI BFS

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09015/top)

Cho đồ thị có hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Sử dụng thuật toán BFS, hãy kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uÎV, vÎV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1  6 9  1 2 2 4 3 1 3 2 3 5 4 3 5 4 5 6 6 4 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

int ma[1005][1005];

void init(){

memset(chuaxet,true,sizeof(chuaxet));

memset(ma,0,sizeof(ma));

}

int DFS(int u){

stack<int> k;

k.push(u);

int truoc[a+5];

memset(truoc,0,sizeof(truoc));

chuaxet[u]=false;

while(!k.empty()){

int top=k.top();

k.pop();

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

if(ma[top][i]==1&&chuaxet[i]){

chuaxet[i]=false;

k.push(top);

k.push(i);

truoc[i]=top;

break;

}

else if(chuaxet[i]==false&&i!=truoc[top]&&ma[top][i]==1&&i==u){

return 1;

}

}

}

return 0;

}

int check(){

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

memset(chuaxet,true,sizeof(chuaxet));

if(DFS(i)) return 1;

}

return 0;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ma[u][v]=1;

// ma[v][u]=1;

}

if(check()) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

return 0;

}

28.  KẾT BẠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09032/top)

Trường học X có N sinh viên, trong đó có M cặp là bạn bè của nhau. Bạn của bạn cũng là bạn, tức là nếu A là bạn của B, B là bạn của C thì A và C cũng là bạn bè của nhau.

Các bạn hãy xác định xem số lượng sinh viên nhiều nhất trong một nhóm bạn là bao nhiêu?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi 2 số nguyên N và M (N, M ≤ 100 000).

M dòng tiếp theo, mỗi dòng gồm 2 số nguyên u, v (u #v) cho biết sinh viên u là bạn của sinh viên v.

**Output:**

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

**Ví dụ:**

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

//by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

//void init(vector<int> res[],bool ok[]){

// for(int i=0;i<=a;i++){

// res[i].clear();

// ok[i]=false;

// }

//}

int BFS(int s,int v,vector<int> res[],bool ok[]){

queue<int> q;

q.push(s);

ok[s]=true;

int t=1;

while(!q.empty()){

s=q.front();q.pop();

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

if(!ok[res[s][i]]){

t=t+1;

q.push(res[s][i]);

ok[res[s][i]]=true;

}

}

}

return t;

}

int slove(int a,vector<int> res[],bool ok[]){

int tmp=0;

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

if(!ok[i]){

int t=BFS(i,a,res,ok);

tmp=(tmp<t)?t:tmp;

}

}

return tmp;

}

int main(){

ios\_base::sync\_with\_stdio(0);cin.tie(0);

int t;cin>>t;

while(t--){

cin>>a>>b;

vector<int> res[a+5];

bool ok[a+5]={};

//init(res,ok);

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

int u,v;cin>>u>>v;

res[u-1].push\_back(v-1);

res[v-1].push\_back(u-1);

}

cout<<slove(a,res,ok)<<endl;

}

return 0;

}

Dạng 10: Đồ Thị Nâng Cao

1. LIỆT KÊ ĐỈNH TRỤ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09018/top)

Cho đồ thị vô hướng liên thông G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy đưa ra tất cả các đỉnh trụ của đồ thị?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra danh sách các đỉnh trụ của mỗi test  theo từng dòng.

**Ví dụ:**

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

// by hung nguyen

// duyet dinh tru

#include<bits/stdc++.h>

using namespace std;

int v, e;

vector<int>vtr[1001];

int check[1001];

void ResetCheck(){

memset(check, 0, sizeof(check));

}

int dinhtru(int u){

queue<int>que;

que.push(u);

check[u]=1;

int count=0;

while(!que.empty()){

int x=que.front();

count++;

que.pop();

for(int i=0; i<vtr[x].size(); i++){

if(check[vtr[x][i]]==0){

que.push(vtr[x][i]);

check[vtr[x][i]]=1;

}

}

}

return count;

}

void ResetVector(){

for(int i=0; i<1001; i++)

vtr[i].clear();

}

int main(){

int T;

cin>>T;

while(T--){

cin>>v>>e;

ResetVector();

ResetCheck();

int x, y;

vector<int> res;

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

cin>>x>>y;

vtr[x].push\_back(y);

vtr[y].push\_back(x);

}

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

check[i]=1;

if(i==1){

int vertex=dinhtru(2);

if(vertex!=v-1) res.push\_back(i);

}

else{

int vertex=dinhtru(1);

if(vertex!=v-1) res.push\_back(i);

}

ResetCheck();

}

sort(res.begin(), res.end());

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

cout<<res[i]<<" ";

cout<<endl;

}

return 0;

}

2. KIỂM TRA CHU TRÌNH

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA09019/top)

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh,  số cạnh của đồ thị; Dòng tiếp theo đưa vào các bộ đôi uV, vV tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 1  6  9  1 2 1 3 2 3 2 5 3 4 3 5 4 5 4 6 5 6 | YES |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int a,b;

bool chuaxet[1005];

int ma[1005][1005];

void init(){

memset(chuaxet,true,sizeof(chuaxet));

memset(ma,0,sizeof(ma));

}

int DFS(int u){

stack<int> k;

k.push(u);

int truoc[a+5];

memset(truoc,0,sizeof(truoc));

chuaxet[u]=false;

while(!k.empty()){

int top=k.top();

k.pop();

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

if(ma[top][i]==1&&chuaxet[i]){

chuaxet[i]=false;

k.push(top);

k.push(i);

truoc[i]=top;

break;

}

else if(chuaxet[i]==false&&i!=truoc[top]&&ma[top][i]==1&&i==u){

return 1;

}

}

}

return 0;

}

int check(){

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

memset(chuaxet,true,sizeof(chuaxet));

if(DFS(i)) return 1;

}

return 0;

}

int main(){

int T;cin>>T;

while(T--){

cin>>a>>b;

init();

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

int u,v;

cin>>u>>v;

ma[u][v]=1;

ma[v][u]=1;

}

if(check()) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

return 0;

}

DẠNG 11: DUYỆT CÂY

1. KIỂM TRA NODE LÁ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11008/top)

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem tất cả các node lá của cây có cùng một mức hay không? Ví dụ với cây dưới đây sẽ cho ta kết quả là Yes.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2 1 2 R 1 3 L 4 10 20 L 10 30 R 20 40 L 20 60 R | 1 0 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7;

struct Node{

int data;

struct Node \*right;

struct Node \*left;

};

Node\* newNode(int data){

Node\* node = new Node();

if(!node){

return NULL;

}

node->data = data;

node->left = NULL;

node->right = NULL;

return node;

}

Node\* insertNode(Node\* root, int data, char x, int v){

if(root == NULL){

root = newNode(data);

}

queue<Node \*> q;

q.push(root);

while (!q.empty()){

Node \*tmp = q.front();

q.pop();

if(tmp->data == data){

if(x == 'L' && tmp->left == NULL){

tmp->left = newNode(v);

return root;

}

else if (x == 'R' && tmp->right == NULL){

tmp->right = newNode(v);

return root;

}

}

else{

if (tmp->left != NULL)

q.push(tmp->left);

if (tmp->right != NULL)

q.push(tmp->right);

}

}

return root;

}

int leafLevel\_node(Node\* node){

if(node == NULL){

return 0;

}

int level = -1;

int leafL = 0;

queue<Node \*> q;

q.push(node);

while(!q.empty()){

leafL++;

int levelP = q.size();

while(levelP > 0){

Node \*node = q.front();

q.pop();

if(node->left != NULL){

q.push(node->left);

}

if(node->right != NULL){

q.push(node->right);

}

if(node->left == NULL && node->right == NULL){

if(level == -1){

level = leafL;

}

else if(level != leafL){

return 0;

}

}

levelP--;

}

}

return 1;

}

void input(){

int u, v;

char x;

int n;

cin >> n;

Node \*root = NULL;

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

cin >> u >> v >> x;

root = insertNode(root, u, x, v);

}

cout << leafLevel\_node(root) << endl;

}

int main(){

int t;

cin >> t;

while(t--){

input();

}

}

2. CÂY BIỂU THỨC 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11001/top)

Cây biểu thức là một cây nhị phân trong đó mỗi node trung gian là một phép toán, mỗi node lá là một toán hạng. Ví dụ với biểu thức P = 3 + ((5+9)\*2) sẽ được biểu diễn như cây dưới đây.  
  
Đối với cây biểu thức, duyệt theo thứ tự trước ta sẽ được biểu thức tiền tố, duyệt theo thứ tự sau ta sẽ được biểu thức hậu tố, duyệt theo thứ tự giữa ta được biểu thức trung tố. Chú ý, cây biểu thức luôn là cây nhị phân đầy (mỗi node trung gian đều có hai node con).

Cho biểu thức hậu tố P, hãy sử dụng cây biểu thức để đưa ra biểu thức trung tố tương ứng với biểu thức P.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test là một biểu thức hậu tố P.
* T, P thỏa mãn ràng buộc : 1≤T≤100; 1≤lengh(P)≤100.

**Output:**

* Đưa ra biểu thức trung tố tương ứng với P.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 ab+ef\*g\*- wlrb+-\* | a + b - e \* f \* g  w \* l - r + b |

// by hung nguyen

#include <bits/stdc++.h>

long long mod = 1e9 + 7;

using namespace std;

struct Node{

char data;

struct Node \*right;

struct Node \*left;

};

Node\* newNode(char data){

Node\* node = new Node();

if(!node){

return NULL;

}

node->data = data;

node->left = NULL;

node->right = NULL;

return node;

}

bool isOperator(char c){

if(c == '+' || c == '-' || c == '\*' || c == '/' || c == '^')

return true;

return false;

}

void InOrder(Node \*node){

if(node){

InOrder(node->left);

cout << node->data;

InOrder(node->right);

}

}

Node\* expression\_tree(string s){

stack<Node \*> st;

Node \*node, \*first, \*second;

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

if(!isOperator(s[i])){

node = newNode(s[i]);

st.push(node);

}

else {

node = newNode(s[i]);

first = NULL;

second = NULL;

if(!st.empty()){

first = st.top();

st.pop();

}

if(!st.empty()){

second = st.top();

st.pop();

}

node->right = first;

node->left = second;

st.push(node);

}

}

node = st.top();

st.pop();

return node;

}

void input(){

string s;

cin >> s;

InOrder(expression\_tree(s));

}

int main(){

int t;

cin >> t;

while(t--){

input();

cout << endl;

}

}

4. DUYỆT CÂY 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11003/top)

Cho phép duyệt cây nhị phân Inorder và Preorder, hãy đưa ra kết quả phép duyệt Postorder của cây nhị phân. Ví dụ với cây nhị phân có các phép duyệt cây nhị phân của cây dưới đây:

Inorder     : 4  2  5  1  3  6

Preorder:  : 1  2  4  5  3  6

Postorder : 4  5  2  6  3  1

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng node; dòng tiếp theo đưa vào N số theo phép duyệt Inorder; dòng cuối cùng đưa vào N số là kết quả của phép duyệt Preorder; các số được viết cách nhau một vài khoảng trống.
* T, N, node thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤1000; 1≤ giá trị node ≤104;

**Output:**

* Đưa ra kết quả phép duyệt Postorder theo từng dòng.

**Ví dụ:**

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

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

int n;

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

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

if (a[i] == x)

return i;

return -1;

}

void init(int in[], int pre[],int n){

int root= tkiem(in, pre[0], n);

if (root != 0)

init(in, pre + 1, root);

if (root != n - 1)

init(in + root + 1, pre + root + 1, n - root - 1);

cout << pre[0] << " ";

}

int main(){

int t;

cin >> t;

while(t--){

cin >> n;

int pre[n];

int ino[n];

for(int i=0;i<n;i++) cin>>ino[i];

for(int i=0;i<n;i++) cin>>pre[i];

init(ino, pre, n);

cout << endl;

}

}

5. DUYỆT CÂY THEO MỨC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11004/top)

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo Level-order. Phép duyệt level-order trên cây là phép thăm node theo từng mức của cây. Ví dụ với cây dưới đây sẽ cho ta kết quả của phép duyệt level-order: 20  8  22  4  12  10  14.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả phép duyệt level-order theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 2 1 2 R 1 3 L 4 10 20 L 10 30 R 20 40 L 20 60 R | 1 3 2  10 20 30 40 60 |

// by hung nguyen

#include <bits/stdc++.h>

using namespace std;

long long mod = 1e9 + 7;

struct Node{

int data;

struct Node \*right;

struct Node \*left;

};

Node\* newNode(int data){

Node\* node = new Node();

if(!node){

return NULL;

}

node->data = data;

node->left = NULL;

node->right = NULL;

return node;

}

Node\* insertNode(Node\* root, int data, char x, int v){

if(root == NULL){

root = newNode(data);

}

queue<Node \*> q;

q.push(root);

while (!q.empty()){

Node \*tmp = q.front();

q.pop();

if(tmp->data == data){

if(x == 'L' && tmp->left == NULL){

tmp->left = newNode(v);

return root;

}

else if (x == 'R' && tmp->right == NULL){

tmp->right = newNode(v);

return root;

}

}

else{

if (tmp->left != NULL)

q.push(tmp->left);

if (tmp->right != NULL)

q.push(tmp->right);

}

}

return root;

}

void lever(Node\* node){

if(node == NULL){

cout << "Node NULL" << endl;

return;

}

queue<Node \*> q;

q.push(node);

while(!q.empty()){

Node \*node = q.front();

q.pop();

if(node->data != 0){

cout<<node->data<<" ";

}

if(node->left != NULL)

q.push(node->left);

if(node->right != NULL)

q.push(node->right);

}

}

void input(){

int u, v;

char x;

int n;

cin >> n;

Node \*root = NULL;

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

cin >> u >> v >> x;

root = insertNode(root, u, x, v);

}

lever(root);

}

int main(){

int t;

cin >> t;

while(t--){

input();

cout << endl;

}

}

10. CÂY NHỊ PHÂN HOÀN HẢO

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11010/top)

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có phải là một cây hoàn hảo hay không (perfect tree)? Một cây nhị phân được gọi là cây hoàn hảo nếu tất cả các node trung gian của nó đều có hai node con và tất cả các node lá đều có cùng một mức.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 6 10 20 L 10 30 R 20 40 L 20 50 R 30 60 L 30 70 R 2 18 15 L 18 30 R 5 1 2 L 2 4 R 1 3 R 3 5 L 3 6 R | Yes Yes No |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int n;

struct node{

int va;

node \*left,\*right;

};

struct data{

int cha,con;

char lr;

};

node\* newNode(int gt){

node \*tmp=new node;

tmp->va=gt;

tmp->left=NULL;

tmp->right=NULL;

return tmp;

}

bool kiemTramuc(node \*tree,int muc,int &mucLa){

if(tree==NULL) return true;

if(tree->left==NULL&&tree->right==NULL){

if(mucLa==0){

mucLa=muc;

return true;

}

else return (muc==mucLa);

}

return(kiemTramuc(tree->left,muc+1,mucLa)&&kiemTramuc(tree->right,muc+1,mucLa));

}

bool kiemTracon(node \*tree){

if(tree==NULL) return true;

if(tree->left==NULL&&tree->right==NULL) return true;

if((tree->left==NULL&&tree->right!=NULL)||(tree->left!=NULL&&tree->right==NULL)) return false;

if(tree->left!=NULL&&tree->right!=NULL)

return (kiemTracon(tree->left)&&kiemTracon(tree->right));

}

void in(node \*tree){

int muc=0,mucLa=0;

if(kiemTracon(tree)==true&&kiemTramuc(tree,muc,mucLa)==true) cout<<"Yes";

else cout<<"No";

}

node\* xaydungTree(data \*a,int n){

node \*tree=NULL;

map<int,node\*> m;

int i=0;

while(i<n){

node \*parent;

if(m.find(a[i].cha)==m.end()){

parent=newNode(a[i].cha);

m[a[i].cha]=parent;

if(tree==NULL) tree=parent;

}

else parent=m[a[i].cha];

node \*child=newNode(a[i].con);

if(a[i].lr=='L') parent->left=child;

else parent->right=child;

m[a[i].con]=child;

i++;

}

return tree;

}

int main(){

int t;

cin >> t;

while(t--){

int n;

cin>>n;

data a[n];

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

cin>>a[i].cha>>a[i].con>>a[i].lr;

}

in(xaydungTree(a,n));

cout<<endl;

}

return 0;

}

11.  CÂY NHỊ PHÂN ĐỦ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11011/top)

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có phải là một cây đủ hay không (full binary tree)? Một cây nhị phân được gọi là cây đủ nếu tất cả các node trung gian của nó đều có hai node con.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4 1 2 L 1 3 R 2 4 L 2 5 R 3 1 2 L 1 3 R 2 4 L | 1 0 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int check;

int n;

struct node{

int data;

node \*l;

node \*r;

node(int x){

data=x;

l=NULL;

r=NULL;

}

};

typedef node NODE;

void noinode(NODE\* &t,int b,char c){

if(c=='L') t->l = new NODE(b);

if(c=='R') t->r = new NODE(b);

}

void taotree(NODE\* t,int a,int b,char c){

if(t==NULL) return;

if(t->data == a) noinode(t,b,c);

taotree(t->l,a,b,c);

taotree(t->r,a,b,c);

}

void KiemTra(NODE \*t){

if(t == NULL || check == 0) return;

if((t->l == NULL && t->r != NULL) || (t->l != NULL && t->r == NULL)) check = 0;

KiemTra(t->l);

KiemTra(t->r);

}

int main(){

int t;

cin>>t;

while(t--){

check = 1;

cin>>n;

NODE\* t = NULL;

while(n--){

int a,b;

char c;

cin>>a>>b>>c;

if(t == NULL){

t = new NODE(a);

noinode(t,b,c);

}

else taotree(t,a,b,c);

}

KiemTra(t);

cout<<check<<endl;

}

}

12. CÂY NHỊ PHÂN BẰNG NHAU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11012/top)

Cho hai cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có giống nhau hay không?

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái của mỗi cây; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 2 1 2 L 1 3 R 2 1 2 L 1 3 R 2 1 2 L 1 3 R 2 1 3 L 1 2 R | 1 0 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

string s1,s2;

int n1,n2;

void solve(){

if(s1==s2) cout<<1<<endl;

else cout<<0<<endl;

}

int main(){

int t;

cin >> t;

while(t--){

cin>>n1;

cin.ignore();

getline(cin,s1);

cin>>n2;

cin.ignore();

getline(cin,s2);

solve();

}

return 0;

}

 13.TỔNG NODE LÁ BÊN PHẢI

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11014/top)

Cho cây nhị phân, nhiệm vụ của bạn là tính tổng của tất cả các node lá bên phải trên cây? Ví dụ với cây dưới đây ta có kết quả là 2.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
* T, N, u, v, thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤103; 1≤u, v≤104;

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 2 1 2 L 1 3 R 5 10 20 L 10 30 R 20 40 L 20 60 R 30 90 L | 3 60 |

// by hung nguyen

#include<iostream>

#include<vector>

#include<map>

using namespace std;

struct Node{

int data;

Node \*left, \*right;

Node(int val){

data = val;

left = right = NULL;

}

};

Node\* xaydungTree(int n){

map<int,Node \*> m;

Node \*root = NULL;

while(n--)

{

int a ,b;

char c;

cin >> a >> b >> c;

Node \*cha;

if(m.find(a)==m.end())

{

cha=new Node(a);

m[a]=cha;

if(root==NULL)

root=cha;

}

else cha=m[a];

Node \*con=new Node(b);

if(c=='L')

cha->left=con;

else if(c=='R')

cha->right=con;

m[b]=con;

}

return root;

}

bool isLeaf(Node \*node){

if(node == NULL){

return false;

}

if(node->left == NULL && node->right == NULL){

return true;

}

return false;

}

long long Sum(Node \*root){

long long res = 0;

if(root){

if(isLeaf(root->right)){

res += root->right->data;

}

else res+= Sum(root->right);

res+= Sum(root->left);

}

return res;

}

int main(){

int t;

cin >> t;

while(t--){

int len;

cin >> len;

Node \*root = xaydungTree(len);

cout << Sum(root) << endl;

}

return 0;

}

19. NODE LÁ CỦA CÂY NHỊ PHÂN TÌM KIẾM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA11021/top)

Cho dãy số gồm N số là phép duyệt theo thứ tự trước (Preorder) của một cây nhị phân tìm kiếm. Hãy in ra tất cả các node lá của cây ?

Ví dụ với dãy A[] = {30, 20, 15, 25, 23, 28, 40, 35, 33, 38, 45} là phép duyệt cây theo thứ tự trước sẽ cho ta kết quả: 15, 23, 28, 33, 38, 45.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T (T≤100).
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên N (N≤1000). Dòng tiếp theo là N số là phép duyệt theo thứ tự trước của cây BST.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  6  10 5 1 7 40 50  11  30 20 15 25 23 28 40 35 33 38 45 | 1 7 50  15 23 28 33 38 45 |

// by hung nguyen

#include<bits/stdc++.h>

using namespace std;

int Binary(int \*x,int l,int r,int ans){

int mid=(l+r)/2;

if(ans==x[mid]) return mid;

if(ans>x[mid]) return Binary(x,mid+1,r,ans);

else return Binary(x,l,mid-1,ans);

}

void nodeLa(int \*x,int \*y,int l,int r,int n,int &index){

if(l==r){

cout<<x[l]<<" ";

index++;return;

}

if(l<0||r>=n||l>r) return;

int tmp=Binary(x,l,r,y[index]);

index++;

nodeLa(x,y,l,tmp-1,n,index);

nodeLa(x,y,tmp+1,r,n,index);

}

void slove(int n,int \*y){

int x[n];

for(int i=0;i<n;i++) x[i]=y[i];

sort(x,x+n);

int index=0;

nodeLa(x,y,0,n-1,n,index);

}

int main(){

int t;

cin >> t;

while(t--){

int n;cin>>n;

int y[n];

for(int i=0;i<n;i++) cin>>y[i];

slove(n,y);

cout<<endl;

}

return 0;

}