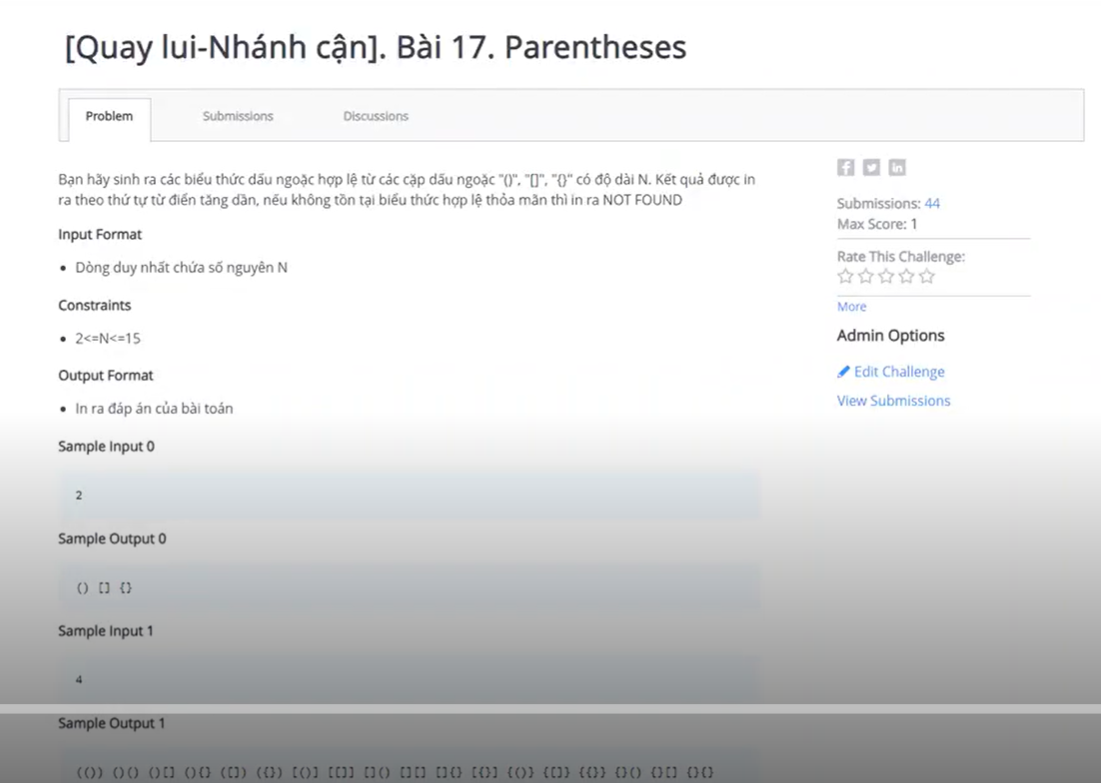
**Bài 1 :**

****

#include <bits/stdc++.h>

using namespace std;

string s="";

string tmp = "[](){}";

int n;

vector<string> res;

bool check(string s){

    stack<char> st;

    for(char x : s){

        if(x=='[' || x=='{' || x =='('){

            st.push(x);

        }

        else{

            if(st.empty()) return false;

            char y = st.top();

            st.pop();

            if((x==')'&&y!='(')||(x==']'&&y!='[')||(x=='}'&&y!='{')){

                return false;

            }

        }

    }

    if(st.empty()) return true;

    return false;

}

void Try(int i){

    for(int j = 0 ; j< tmp.size();j++){

        s+=tmp[j];

        if(s.size()==n){

        //    if(check(s))

            res.push\_back(s);

        }

        else Try(i+1);

        s.pop\_back();

    }

}

int main(){

    cin >> n;

    if(n%2==1){

        cout << "NOT FOUND" << endl;

    }

    else{

        Try(1);

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

        for(auto x : res){

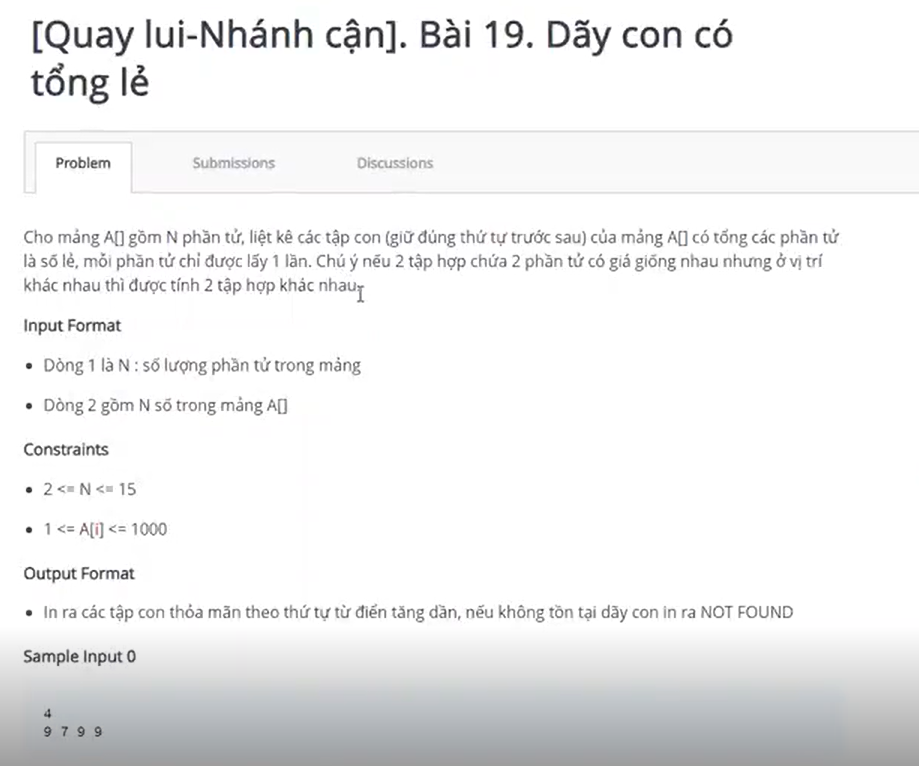
            cout << x <<" ";

        }

    }

}

**Bài 2 :**

****

****

#include <bits/stdc++.h>

using namespace std;

int n , a[100] , X[100];

vector <vector<int>> res;

void nhap(){

    cin >> n;

    for(int i = 0  ; i < n;i++){

        cin >> a[i];

    }

}

void  Try(int i , int idx , int sum){

    for(int j = idx ; j < n ; j++){

        sum+=a[j];

        X[i]=a[j];

        if(sum%2==1){

            vector<int> tmp(X+1,X+i+1);

            res.push\_back(tmp);

        }

        Try(i+1,j+1,sum);

        sum-=a[j];

    }

}

// Tránh lặp chỉ tính 1 lần ko tính thứ tự vị trí thì j+1 còn lặp mà tính cả hoán vị thì dùng idx+1

int main(){

    nhap();

    Try(1,0,0);

    //sort(res.begin(),res.end());

    for(vector<int> tmp : res){

        for(int x : tmp){

            cout << x <<" ";

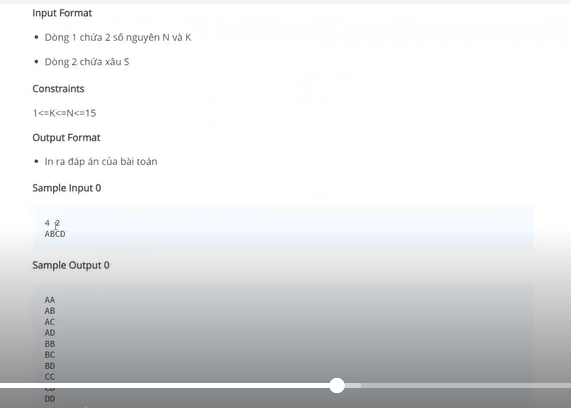
        }

        cout << endl;

    }

}

**Bài 3 :**

****

#include <bits/stdc++.h>

using namespace std;

int n , k;

string s;

void nhap(){

    cin >> n >> k;

    cin >> s;

    sort(s.begin(),s.end());

}

string tmp="";

vector<string> res;

void Try(int idx){

    for(int j = idx ; j < n ; j++){

        tmp += s[j];

        if(tmp.size()==k){

            cout << tmp << endl;

        }

        else{

            Try(j);

        }

        tmp.pop\_back();

    }

}

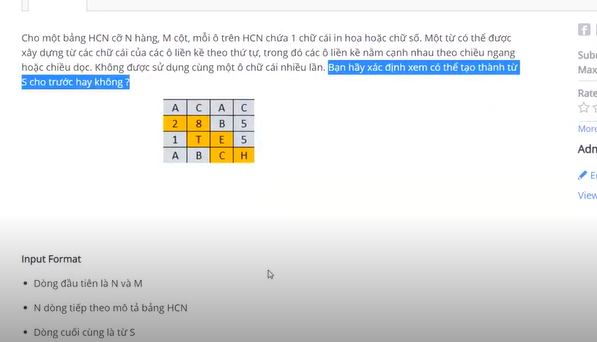
int main(){

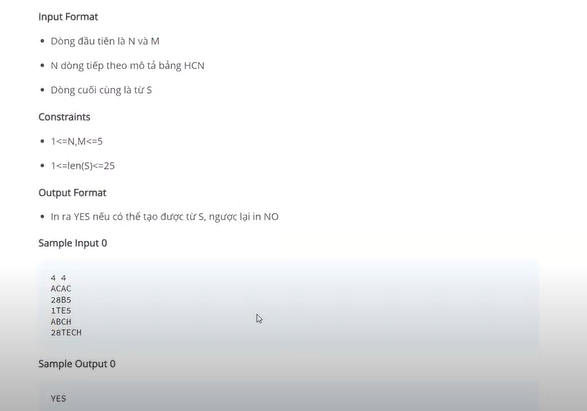
    nhap();

    Try(0);

}

**Bai 4 :**

****

****

#include <bits/stdc++.h>

using namespace std;

int n , m;

char a[100][100];

string s , tmp="";

bool visited[100][100];

int ok =0 ;

int dx[4]= {-1 , 0 , 0 , 1};

int dy[4]= {0 , -1 , 1 , 0};

void nhap(){

    memset(visited,false,sizeof(visited));

    cin >> n >> m;

    for(int i = 1 ; i <= n ; i++){

        for(int j = 1 ;  j<=m ;j++){

            cin >> a[i][j];

        }

    }

    cin >> s;

}

void Try(int i , int j){

    if(ok) return;

    if(tmp==s){

        ok = 1;

        return;

    }

    for(int k = 0 ; k < 4 ; k++){

        int i1 = i + dx[k];

        int j1 = j + dy[k];

        if( i1 >= 1 && i1 <= n && j1 >= 1 && j1 <= n  && !visited[i1][j1]){

            tmp += a[i1][j1];

            visited[i1][j1]=true;

            if(tmp == s){

                ok =1 ;

            }

            else if( s.find(tmp) != string :: npos){ // tim vi tri tmp trong s

                Try(i1,j1);

            }

            tmp.pop\_back();

            visited[i1][j1]=false;

        }

    }

}

int main(){

    nhap();

    for(int i = 1; i<=n ;i++){

        for(int j = 1 ;  j<= m ;j++){

            if(a[i][j]==s[0]){

                memset(visited,false,sizeof(visited));

                tmp = "";

                tmp += a[i][j];

                visited[i][j]=true;

                Try(i,j);

                if(ok){

                    cout << "YES" << endl;

                    return 0;

                }

            }

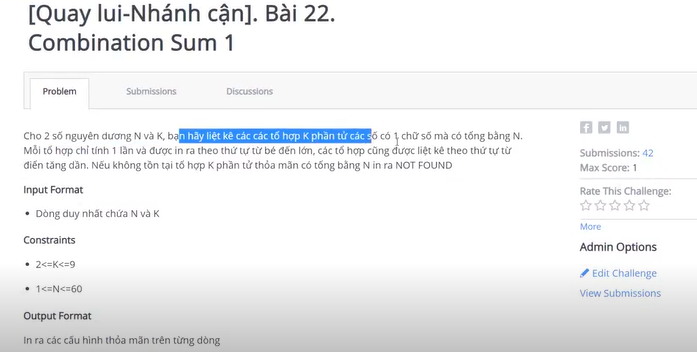
        }

    }

    cout << "NO" << endl;

}

**Bai 5 :**

****

****

#include <bits/stdc++.h>

using namespace std;

int n , k;

vector<vector<int>> res;

vector <int> tmp;

void Try(int init , int sum){

    for(int j = init ; j <= 9 ; j++){

        tmp.push\_back(j);

        sum+=j;

        if(sum==n&&tmp.size()==k){

            res.push\_back(tmp);

        }

        else if(sum < n && tmp.size()<k){

            Try(j+1,sum);

        }

        tmp.pop\_back();

        sum-=j;

    }

}

int main(){

    cin >>  n >> k;

    Try(1,0);

    if(res.size()==0){

        cout << "NOT FOUND" << endl;

    }

    else{

        sort(res.begin(),res.end());

        for(vector <int> tmp : res){

            for(int j = 0 ; j < tmp.size() ; j++){

                cout << tmp[j];

                if(j!= tmp.size()-1) cout <<" + ";

            }

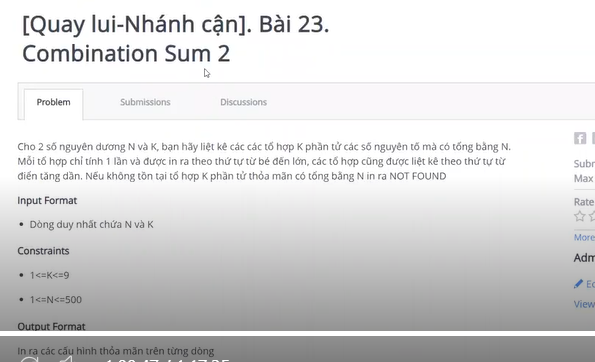
            cout << endl;

        }

    }

}

**Bai 6 :**

****

****

#include <bits/stdc++.h>

using namespace std;

int n , k;

vector<vector<int>> res;

vector <int> tmp;

vector <int> prime;

bool nt(int a ){

    if(a==0&&a==1) return false;

    if(a==2) return true;

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

        if(a%i==0)  return false;

    }

    return true;

}

void Try(int init , int sum){

    for(int j = init ; j <= prime.size() ; j++){

        tmp.push\_back(prime[j]);

        sum+=prime[j];

        if(sum==n&&tmp.size()==k){

            res.push\_back(tmp);

        }

        else if(sum < n && tmp.size()<k){

            Try(j+1,sum);

        }

        tmp.pop\_back();

        sum-=prime[j];

    }

}

int main(){

    cin >>  n >> k;

    for(int i = 1 ; i<=500 ;i++){

        if(nt(i)) prime.push\_back(i);

    }

    Try(0,0);

    if(res.size()==0){

        cout << "NOT FOUND" << endl;

    }

    else{

        sort(res.begin(),res.end());

        for(vector <int> tmp : res){

            for(int j = 0 ; j < tmp.size() ; j++){

                cout << tmp[j];

                if(j!= tmp.size()-1) cout <<" + ";

            }

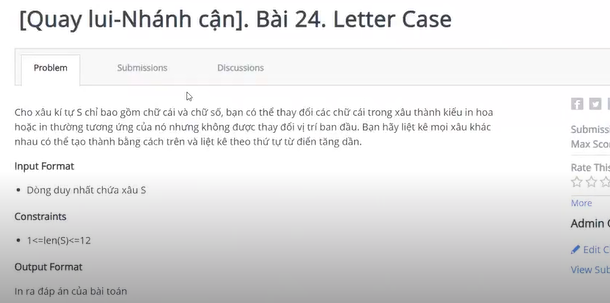
            cout << endl;

        }

    }

}

**Bai 7 :**

****

****

#include <bits/stdc++.h>

using namespace std;

string s;

set<string> res;

string tmp;

void Try(int idx){

    for(char x : {tolower(s[idx]),toupper(s[idx])}){

        tmp += x;

        if(tmp.size()==s.size()){

            res.insert(tmp);

        }

        else{

            Try(idx+1);

        }

        tmp.pop\_back();

    }

}

int main(){

    cin>>s;

    Try(0);

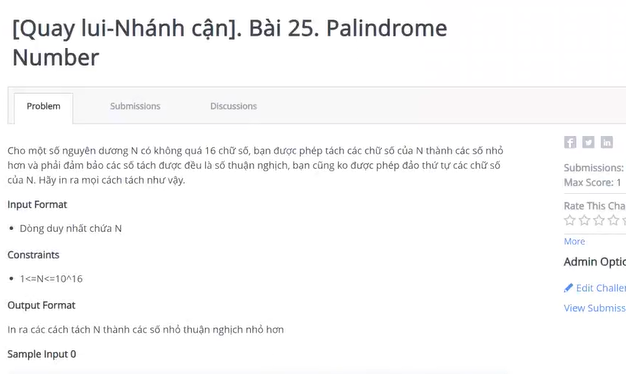
    for(string x : res){

        cout << x << endl;

    }

}

**Bai 8 :**

****

****

#include <bits/stdc++.h>

using namespace std;

bool check(string s){

    int l = 0 ;

    int r = s.size()-1;

    while(l<r){

        if(s[l]!=s[r]) return false;

        l++;

        r--;

    }

    return true;

}

vector<vector<string>> res;

vector<string> v;

void Try(int start , string s){

    if(start >= s.size()) res.push\_back(v);

    for(int end = start ; end < s.size() ; end++){

        if(check(s.substr(start , end - start +1))){

            v.push\_back(s.substr(start,end-start+1));

            Try(end+1,s);

            v.pop\_back();

        }

    }

}

int main(){

    string s;

    cin >> s;

    Try(0,s);

    for(auto it : res){

        for(string x : it ) cout  << x << " ";

        cout << endl;

    }

}