Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	6	0	346

Task A ()

```
n = int(input())
print(n-1)
```

Task B ()

```
import sys
import math
eps = 0.001
class vector:
    a = 0
    b = 0
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def __add__(self, other):

        return vector(self.a + other.a, self.b+ other.b)
    def __sub__(self, other):
        return vector(self.a - other.a, self.b- other.b)
    def __str__(self):
        return str(self.a) +" "+str(self.b)
    def __mul__(self, other):
        return (self.a*other.a+self.b*other.b)
    def sizeof(self):
        return math.sqrt(self.a*self.a + self.b*self.b)
n = int(input())
if(n==6):
    arr = []
    a,b, c  =0, 0, 0
    for i in range(6):
        arr.append(list(map(float, input().split())))
    for i1 in range(6):
        for i2 in range(6):
            for i3 in range(6):
                if(i1 ==i2 or i2==i3 or i3==i1):
                    continue
                u1 = vector(arr[i1][0], arr[i1][1])
                u2 = vector(arr[i2][0], arr[i2][1])
                u3 = vector(arr[i3][0], arr[i3][1])
                cos = (u3-u2)*(u3-u1)
                cos/=vector.sizeof((u3-u2))
                cos/=vector.sizeof((u3-u1))
                #print(cos)
                if(abs(cos +1/2)<eps    ):
                    a = u1
                    b = u2
                    c = u3

    print(a)
    print(c)
    print(b)


else:
    arr = []
    for i in range(3):
        arr.append(list(map(float, input().split())))
    b = vector(arr[1][0], arr[1][1])
    c = vector(arr[2][0], arr[2][1])
    a = vector(arr[0][0], arr[0][1])
    for i in range(6):
        print(a)
        a, b, c = b, c, c+c+a -b-b
```

Task C ()

```
#include <iostream>
#include<vector>
#include<string>
#include<cmath>
using namespace std;
int gc(char x){
    return x-'a';
}

int main() {
    string s;
    cin>>s;
    int n;
    cin>>n;
    int totans = 0;
    vector<vector<int>> poses(26);
    for(int i = 0;i< s.size();i++){
        poses[gc(s[i])].push_back(i);
    }
    for(int zxc = 0;zxc< n;zxc++){
        string s1;
        cin>>s1;
        int ans = s.size();
        for(int i=0;i<s1.size();i++){
            char nxc = gc(s1[i]);
            if(0 <= poses[nxc].size()){
                vector<int>index(26, 0);
                int cid = -1;
                int num = 0;
                int u = i;
                while(u<s1.size()){
                    while(index[gc(s1[u])]< poses[gc(s1[u])].size() && poses[gc(s1[u])][index[gc(
                        s1[u])]] <= cid){
                        index[gc(s1[u])]++;
                    }
                    if(index[gc(s1[u])]>= poses[gc(s1[u])].size()){
                        break;
                    }
                    num++;
                    cid = poses[gc(s1[u])][index[gc(s1[u])]];
                    u++;
                }
                if(s.size()-num < ans){
                    ans = s.size()-num;
                }
            }
        }
        totans+=ans;

    }
    cout<<totans;
    return 0;
}
```

Task D ()

```
#include<iostream>
#include<vector>
#include<set>
#include<cmath>
    using namespace std;
    bool isin(int x, int y, int n, int m){
        return x<n && y<m && y>=0 && x>=0;
    }
    int gt(int x, int y, int m){
        return x*m + y;
    }
    int main(){
        int n, m;
        cin>>n>>m;
        int sx, sy, fx, fy;
        cin>>sx>>sy>>fx>>fy;
        fx--;fy--;sx--;sy--;
        vector<vector<pair<int, int>>> field(n, vector<pair<int, int>>(m));
        int INF = 1e9;
        for(int i = 0;i<n;i++){
            for(int j = 0;j< m;j++){
                int a, b;
                cin>>a>>b;
                field[i][j] = pair<int, int>(a, b);
            }
        }
        vector<vector<pair<int, int>>> gr(n*m);
        for(int i = 0;i< n;i++){
            for(int j = 0;j< m;j++){
                for(int x = -i; x<n-i+1;x++){
                    for(int y = -j;y<m-j+1;y++){
                        if(x==0 && y==0){
                            continue;
                        }
                        if(isin(x+i, y+j, n, m)){
                            gr[gt(i, j, m)].push_back(pair<int, int>(gt(i+x, j+y, m), abs(x-field[
                                i][j].first)+ abs(y-field[i][j].second)));
                        }
                    }
                }
            }
        }
        vector<bool> used(n*m, false);
        vector<int>d(n*m, INF);
        d[gt(sx, sy, m)] = 0;
        set<pair<int, int>> heap;
        for(int i = 0;i< n*m;i++){
            heap.insert(pair<int, int>(d[i], i));
        }
        for(int i = 0;i< n*m;i++){
            auto nxt = *heap.begin();
            heap.erase(nxt);
            while(used[nxt.second]){
                nxt = *heap.begin();
                heap.erase(nxt);
            }
            for(auto u: gr[nxt.second]){
                d[u.first] = min(d[u.first], d[nxt.second] + u.second);
                heap.insert(pair<int, int>(d[u.first], u.first));
            }
            used[nxt.second] = true;
        }
        cout<<d[gt(fx, fy, m)];
        return 0;
    }
```

Task E ()

```
coords = [0, 100, 200, 300]
n, m, b = map(int, input().split())
cb = []
for i in range(b):
    cb.append(list(map(int, input().split())))
if(b==1):
    print("?", cb[0][0], cb[0][1], cb[0][0]+coords[1], cb[0][1])
    si, ti = map(int, input().split())
    if(si == cb[0][0]):
        print("!", coords[1]+1, 1)
    else:
        print("!", 1, 1)
```

Task F ()