Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	63	0	403

Task A ()

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

Task B ()

```
k = int(input())
A = []
def S(A,B):
    xa,ya = A
    xb,yb = B
    x = xb + (xb-xa)
    y = yb +(yb-ya)
    return [x,y]
def distance(A,B):
    xa,ya = A
    xb,yb = B
    return ((xa-xb)**2 + (ya-yb)**2)
for i in range(k):
    A.append(list(map(float,input().split())))
if k == 6:
    center = [sum([A[i][0] for i in range(k)])/k,sum([A[i][1] for i in range(k)])/k]
    print(*center)
    print(*A[0])
    D = [10**10] *5
    for i in range(5):
        D[i] = distance(A[0],A[i+1])
    y = min(D)
    for i in range(5):
        if D[i] == y:
            print(*A[i+1])
            break
else:
    Res = [0]*6
    Res[0] = A[1]
    Res[1] = A[2]
    Res[3] = S(A[1],A[0])
    Res[4] = S(A[2],A[0])
    Res[2] = [A[0][0] + A[2][0]-A[1][0],A[0][1] + A[2][1] - A[1][1]]
    Res[5] = S(Res[2],A[0])
    for i in Res:
        print(*i)
```

Task C ()

```cpp
#include <iostream>
#include<set>
#include<vector>
#include<string>
#include<cmath>
using namespace std;

int main()
{
    string t ;
    cin >> t;
    set<int> used;
    for (int i=0;i<t.length();i++){
        used.insert(t[i] - 'a');
    }
    int n;
    cin >> n;
    int ans = 0;
    for(int i = 0;i<n;i++){
        string A;
        cin>> A;
        int best = pow(10,10);
        for(int j = 0;j < A.length();j++){
            int k = t.length();
            int e = j;
            int cur = 0;
            int res = 0;
            while(k > 0 and cur < t.length() and e< A.length()){
                if (used.find(A[e] - 'a') == used.end()){
                    break;
                }
                else{
                    while(t[cur] != A[e]){
                        cur ++;
                        if (cur == t.length()){
                            break;
                        }
                    }
                    cur ++;
                    e++;
                    if (cur <= t.length()){
                        res++;
                        k = k-1;
                    }
                }
            }
            if (best > t.length() - res){
                best = t.length() - res;
            }
        }
        ans  = ans + best;
    }
    cout<<ans<<endl;
    return 0;
}
```

Task D ()

```
#include <iostream>
#include<vector>
#include<algorithm>
#include<cmath>
using namespace std;
bool f(const pair<int,int> a,const pair<int,int> b){
    return (a.first>b.first);
}
vector<vector<int> > Map;
int dijkstra(int u,int v){
    vector<pair<int,int> > que;
    que.push_back(make_pair(0,u));
    vector<bool> Visited;
    for(int i =0;i<Map.size();i++){
        Visited.push_back(false);
    }
    while(que.size()>0){
        pair<int,int> qwe = que[0];
        pop_heap(que.begin(),que.end(),f);
        que.pop_back();
        int dist,k;
        dist = qwe.first;
        k = qwe.second;
        if(Visited[k] == true){
            continue;
        }
        Visited[k] = true;
        if(k == v){
            return dist;
        }
        for(int i = 0; i<Map[k].size();i++){
            if (Visited[i] == false){
                que.push_back(make_pair(dist+Map[k][i],i));
                push_heap(que.begin(),que.end(),f);
            }
        }
    }

}
int main()
{
    int n,m,ar,br,ac,bc;
    cin >> n>>m;
    cin >> ar>>br>>ac>>bc;
    ar = ar-1;
    br = br-1;
    ac = ac-1;
    bc = bc -1;
    for (int i = 0;i< n*m;i++){
        vector<int> B;
        for(int j = 0;j<n*m;j++){
            B.push_back(pow(10,10));
        }
        Map.push_back(B);
    }
    for(int i = 0;i<n;i++){
        vector<int> A;
        for (int qwe = 0;qwe < 2*m;qwe++){
            int pl;
            cin >> pl;
            A.push_back(pl);
        }
        for(int j = 0;j<A.size()/2;j++){
            for(int k = 0;k<n*m;k++){
                int x = A[2*j];
                int y = A[2*j + 1];
                Map[i*m + j][k] = abs(k/m -i - x) + abs(k%m - y - j);
            }
        }
    }
    cout<<dijkstra(ar*m+br,ac*m+bc)<<endl;
}
```

Task E ()

```
import sys
n,m,b = map(int,input().split())
B = []
for i in range(b):
    B.append(list(map(int,input().split())))
A = [i for i in range(2**b)]
Map = [0 for i in range(2**b)]
cur = 0
Len = n+1
Deleting = set()
while True:
    if  Map[A[cur]] == len(B) :
        print('!',Len*A[cur]+1,1)
        break
    r1 = [B[Map[A[cur]]][0]+ Len*A[cur],B[Map[A[cur]]][1]]
    Map[A[cur]] += 1
    cur  = cur+1
    if cur >= len(A):
        cur = 0
        for k in Deleting:
            if k in A:
                A.remove(k)
        Deleting = set()
    while A[cur] in Deleting:
        cur += 1
        if cur >= len(A):
            cur = 0
            for k in Deleting:
                if k in A:
                    A.remove(k)
            Deleting = set()
            break
    if  Map[A[cur]] == len(B):
        print('!',Len*A[cur]+1,1)
        break
    r2 = [B[Map[A[cur]]][0]+ Len*A[cur],B[Map[A[cur]]][1]]
    Map[A[cur]] += 1
    print("?",r1[0],r1[1],r2[0],r2[1])
    sys.stdout.flush()
    p,q = map(int,input().split())
    p = (p-1)//Len
    if p not in A:
        for qwe in A:
            if qwe not in Deleting:
                p =qwe
                break
    Deleting.add(p)
    cur  = cur+1
    if cur >= len(A):
        cur = 0
        for k in Deleting:
            if k in A:
                A.remove(k)
        Deleting = set()
    while A[cur] in Deleting:
        cur += 1
        if cur >= len(A):
            cur = 0
            for k in Deleting:
                if k in A:
                    A.remove(k)
            Deleting = set()
            break
```

Task F ()

```
n,m = map(int,input().split())
if n <= 5:
    if n == 2:
        A = [1]
    elif n == 3:
        A = [0,0,0,3]
    elif n == 4:
        A = [0,0,0,0,0,0,0,0,4,12]
    else:
        A= [0]*20
        A[11] = 5
        A[16] =20
        A[19] = 5*4*3
    for i in range(m):
        print(A[i],end = '␣')

    print('')
else:
    print(1)
```