Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	0	0	320

Task A ()

```
#include <iostream>
#include <queue>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <cmath>
#include <algorithm>
#include <string>
#include <stack>

using namespace std;

/*vector <int> t;

int sum(int v, vector <int> &t, int l, int r, int tl, int tr)
{
        if (l > r)
                return 0;
        if (tl == l && tr == r)
                return t[v];
        int m = (l + r) / 2;
        return sum(v * 2, t, l, min(m, r), tl, m) + sum(v * 2 + 1, t, max(l, m), r, m + 1, tr);
}

int build(int v, vector <int> &t, vector <int> &a, int l, int r)
{
        if (l == r)
        {
                t[v] = a[l];
        }
        else
        {
                int m = (l + r) / 2;
                build(v * 2, t, a, l, m);
                build(v * 2 + 1, t, m + 1, r);
                t[v] = t[v * 2] + t[v * 2 + 1];
        }
}*/

int main()
{
        /*auto cmp = [](int a, int b) { return a < b;};
        priority_queue <int, vector <int>, decltype(cmp)> q(cmp);*/
        /*10
        5 5
        5 0
        8 7
        7 0
        9 8
        8 0
        9 9
        12
        6 6
        6 0
        9 9
```

```
        9 0
        10 11
        10 0
        11 11
        11 0
        9
        4 5
        4 0
        7 6
        6 0
        8 7
        7 0
        8 8
        8 0
        16
        8 8
        8 0
        12 12
        12 0
        14 14
        14 0
        15 15
        15 0
        17
        8 9
        8 0
        12 13
        12 0
        15 14
        14 0
        16 15
        15 0
        16 16*/
        int n;
        cin >> n;
        cout << n - 1;
}
```

Task B ()

```
#include <iostream>
#include <queue>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <cmath>
#include <algorithm>
#include <string>
#include <stack>

using namespace std;

double eps = 0.0001;

/*vector <int> t;

int sum(int v, vector <int> &t, int l, int r, int tl, int tr)
{
        if (l > r)
                return 0;
        if (tl == l && tr == r)
                return t[v];
        int m = (l + r) / 2;
        return sum(v * 2, t, l, min(m, r), tl, m) + sum(v * 2 + 1, t, max(l, m), r, m + 1, tr);
}

int build(int v, vector <int> &t, vector <int> &a, int l, int r)
{
        if (l == r)
        {
                t[v] = a[l];
        }
        else
        {
                int m = (l + r) / 2;
                build(v * 2, t, a, l, m);
                build(v * 2 + 1, t, m + 1, r);
                t[v] = t[v * 2] + t[v * 2 + 1];
        }
}*/

struct v
{
        double x;
        double y;
        double len;
        v(double x1, double x2, double y1, double y2)
        {
                x = x2 - x1;
                y = y2 - y1;
                len = sqrt(x * x + y * y);
        }
};

struct vt
{
        double x;
        double y;
        int id;
};

double scal(v v1, v v2)
{
        return v1.x * v2.x + v1.y * v2.y;
}

double pscal(vt v1, vt v2)
{
        return v1.x * v2.y - v1.y * v2.x;
}
```

```
bool cmp(vt v1, vt v2)
{
	double t = pscal(v1, v2);
	if (t > 0)
		return false;
	else
		return true;
}

int main()
{
	/*auto cmp = [](int a, int b) { return a < b;};
	priority_queue <int, vector <int>, decltype(cmp)> q(cmp);*/
	int n;
	cin >> n;
	vector <pair<double, double>> a(6);
	if (n == 6)
	{
		vector <vt> at(5);
		for (int i = 0; i < 6; i++)
		{
			cin >> a[i].first >> a[i].second;
			if (i > 0)
			{
				at[i - 1].x = a[i].first - a[0].first;
				at[i - 1].y = a[i].second - a[0].second;
				at[i - 1].id = i;
			}
		}
		sort(at.begin(), at.end(), cmp);
		for (int i = 1; i >= 0; i--)
		{
			cout << a[at[i].id].first << " " << a[at[i].id].second << endl;
		}
		cout << a[0].first << " " << a[0].second << endl;
	}
	else
	{
		for (int i = 0; i < 3; i++)
			cin >> a[i].first >> a[i].second;
		v v1(a[1].first, a[0].first, a[1].second, a[0].second), v2(a[1].first, a[2].first,
		    a[1].second, a[2].second);
		v v3(a[1].second, a[0].second, a[1].first, a[0].first);
		v3.y = -v3.y;
		if (scal(v2, v3) < 0)
		{
			v3.x = -v3.x;
			v3.y = -v3.y;
		}
		v3.x = v3.x  * sqrt(3);
		v3.y = v3.y * sqrt(3);
		a[3].first = a[1].first + v3.x;
		a[3].second = a[1].second + v3.y;
		a[4].first = a[0].first + v3.x;
		a[4].second = a[0].second + v3.y;
		v v4(a[2].second, a[3].second, a[2].first, a[3].first);
		v4.y = -v4.y;
		v v5(a[2].first, a[1].first, a[2].second, a[1].second);
		if (scal(v4, v5))
		{
			v4.x = -v4.x;
			v4.y = -v4.y;
		}
		v4.x = v4.x * sqrt(3);
		v4.y = v4.y * sqrt(3);
		a[5].first = a[3].first + v4.x;
		a[5].second = a[3].second + v4.y;
		for (int i = 0; i < 6; i++)
			cout << a[i].first << " " << a[i].second << endl;
	}
}
```

Task C ()

```
#include <iostream>
#include <queue>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <cmath>
#include <algorithm>
#include <string>
#include <stack>

using namespace std;

/*vector <int> t;

int sum(int v, vector <int> &t, int l, int r, int tl, int tr)
{
	if (l > r)
		return 0;
	if (tl == l && tr == r)
		return t[v];
	int m = (l + r) / 2;
	return sum(v * 2, t, l, min(m, r), tl, m) + sum(v * 2 + 1, t, max(l, m), r, m + 1, tr);
}

int build(int v, vector <int> &t, vector <int> &a, int l, int r)
{
	if (l == r)
	{
		t[v] = a[l];
	}
	else
	{
		int m = (l + r) / 2;
		build(v * 2, t, a, l, m);
		build(v * 2 + 1, t, m + 1, r);
		t[v] = t[v * 2] + t[v * 2 + 1];
	}
}*/

int inf = 1e8;

int main()
{
	/*auto cmp = [](int a, int b) { return a < b;};
	priority_queue <int, vector <int>, decltype(cmp)> q(cmp);*/
	string t;
	cin >> t;
	int tlen = t.length();
	vector <int> temp(26, -1);
	vector <int[26]> next(t.length());
	for (int i = t.length() - 1; i >= 0; i--)
	{
		for (int j = 0; j < 26; j++)
			next[i][j] = temp[j];
		temp[t[i] - 'a'] = i;
	}
	int n, ans = 0;
	cin >> n;
	for (int l = 0; l < n; l++)
	{
		string s;
		cin >> s;
		vector <vector <int>> dp(s.length(), vector <int>(t.length(), inf));
		int tans = t.length();
		for (int i = 0; i < s.length(); i++)
		{
			int c = s[i] - 'a';
			if (temp[c] != -1)
				dp[i][temp[c]] = min(temp[c], dp[i][temp[c]]);
			for (int j = 0; j < t.length(); j++)
			{
```

```
                                if (dp[i][j] == inf)
                                        continue;
                                tans = min(tans, dp[i][j] + tlen - j - 1);
                                if (i == s.length() - 1)
                                        continue;
                                int nextc = s[i + 1] - 'a';
                                if (next[j][nextc] != -1)
                                        dp[i + 1][next[j][nextc]] = min(dp[i + 1][next[j][nextc]],
                                             dp[i][j] + next[j][nextc] - j - 1);
                        }
                }
                ans += tans;
        }
        cout << ans;
}
```

Task D ()

```
#include <iostream>
#include <queue>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <cmath>
#include <algorithm>
#include <string>
#include <stack>
#define int long long

using namespace std;

int inf = 1e9;

pair<int, int> a[1000][1000];
int d[1000][1000][9];

struct heh
{
        int i, j, cost, flag;
};

signed main()
{
        heh temp;
        auto cmp = [](heh a, heh b) { return a.cost > b.cost;};
        priority_queue <heh, vector <heh>, decltype(cmp)> q(cmp);
        int n, m;
        cin >> n >> m;
        int si, sj, ei, ej;
        cin >> si >> sj >> ei >> ej;
        si--;
        sj--;
        ei--;
        ej--;
        for (int i = 0; i < 1000; i++)
                for (int j = 0; j < 1000; j++)
                        for (int l = 0; l < 9; l++)
        d[i][j][l] = inf;
        for (int i = 0; i < n; i++)
                for (int j = 0; j < m; j++)
                        cin >> a[i][j].first >> a[i][j].second;
        temp.i = si;
        temp.j = sj;
        temp.cost = 0;
        temp.flag = 0;
        d[si][sj][0] = 0;
        q.push(temp);
        while (!q.empty())
        {
                heh now = q.top();
                q.pop();
                if (now.cost > d[now.i][now.j][now.flag])
                        continue;
                int vi, vj;
                if (now.i == ei && now.j == ej)
                {
                        cout << now.cost;
                        return 0;
                }
                if (now.flag == 0)
                {
                        vi = a[now.i][now.j].first;
                        vj = a[now.i][now.j].second;

                        int need = abs(-1 - vi) + abs(-1 - vj);
                        if ((now.i - 1 >= 0) && (now.j - 1 >= 0))
                        {
        if (now.cost + need < d[now.i - 1][now.j - 1][0])
```

```
{
        temp.cost = now.cost + need;
        temp.i = now.i - 1;
        temp.j = now.j - 1;
        temp.flag = 0;
        q.push(temp);
        d[now.i - 1][now.j - 1][0] = now.cost + need;
}
if (now.cost + need < d[now.i - 1][now.j - 1][2])
{
        temp.cost = now.cost + need;
        temp.i = now.i - 1;
        temp.j = now.j - 1;
        temp.flag = 2;
        q.push(temp);
        d[now.i - 1][now.j - 1][2] = now.cost + need;
}
                }

                need = abs(vi) + abs(-1 - vj);
                if ((now.j - 1 >= 0))
                {
if (now.cost + need < d[now.i][now.j - 1][0])
{
        temp.cost = now.cost + need;
        temp.i = now.i;
        temp.j = now.j - 1;
        temp.flag = 0;
        q.push(temp);
        d[now.i][now.j - 1][0] = now.cost + need;
}
if (now.cost + need < d[now.i][now.j - 1][3])
{
        temp.cost = now.cost + need;
        temp.i = now.i;
        temp.j = now.j - 1;
        temp.flag = 3;
        q.push(temp);
        d[now.i][now.j - 1][3] = now.cost + need;
}
                }

                need = abs(1 - vi) + abs(-1 - vj);
                if ((now.i + 1 < n) && (now.j - 1 >= 0))
                {
if (now.cost + need < d[now.i + 1][now.j - 1][0])
{
        temp.cost = now.cost + need;
        temp.i = now.i + 1;
        temp.j = now.j - 1;
        temp.flag = 0;
        q.push(temp);
        d[now.i + 1][now.j - 1][0] = now.cost + need;
}
if (now.cost + need < d[now.i + 1][now.j - 1][4])
{
        temp.cost = now.cost + need;
        temp.i = now.i + 1;
        temp.j = now.j - 1;
        temp.flag = 4;
        q.push(temp);
        d[now.i + 1][now.j - 1][4] = now.cost + need;
}
                }
                //niz
                need = abs(1 - vi) + abs(vj);
                if ((now.i + 1 < n))
                {
if (now.cost + need < d[now.i + 1][now.j][0])
{
        temp.cost = now.cost + need;
        temp.i = now.i + 1;
        temp.j = now.j;
        temp.flag = 0;
```

```
                q.push(temp);
                d[now.i + 1][now.j][0] = now.cost + need;
        }
        if (now.cost + need < d[now.i + 1][now.j][5])
        {
                temp.cost = now.cost + need;
                temp.i = now.i + 1;
                temp.j = now.j;
                temp.flag = 5;
                q.push(temp);
                d[now.i + 1][now.j][5] = now.cost + need;
        }
                        }
                        //pravo-niz
                        need = abs(1 - vi) + abs(1 - vj);
                        if ((now.i + 1 < n) && (now.j + 1 < m))
                        {
        if (now.cost + need < d[now.i + 1][now.j + 1][0])
        {
                temp.cost = now.cost + need;
                temp.i = now.i + 1;
                temp.j = now.j + 1;
                temp.flag = 0;
                q.push(temp);
                d[now.i + 1][now.j + 1][0] = now.cost + need;
        }
        if (now.cost + need < d[now.i + 1][now.j + 1][6])
        {
                temp.cost = now.cost + need;
                temp.i = now.i + 1;
                temp.j = now.j + 1;
                temp.flag = 6;
                q.push(temp);
                d[now.i + 1][now.j + 1][6] = now.cost + need;
        }
                        }
                        //pravo
                        need = abs(vi) + abs(1 - vj);
                        if ((now.j + 1 < m))
                        {
        if (now.cost + need < d[now.i][now.j + 1][0])
        {
                temp.cost = now.cost + need;
                temp.i = now.i;
                temp.j = now.j + 1;
                temp.flag = 0;
                q.push(temp);
                d[now.i][now.j + 1][0] = now.cost + need;
        }
        if (now.cost + need < d[now.i][now.j + 1][7])
        {
                temp.cost = now.cost + need;
                temp.i = now.i;
                temp.j = now.j + 1;
                temp.flag = 7;
                q.push(temp);
                d[now.i][now.j + 1][7] = now.cost + need;
        }
                        }
                        //pravo-verh
                        need = abs(-1 - vi) + abs(1 - vj);
                        if ((now.i - 1 >= 0)&&(now.j + 1 < m))
                        {
        if (now.cost + need < d[now.i - 1][now.j + 1][0])
        {
                temp.cost = now.cost + need;
                temp.i = now.i - 1;
                temp.j = now.j + 1;
                temp.flag = 0;
                q.push(temp);
                d[now.i - 1][now.j + 1][0] = now.cost + need;
        }
        if (now.cost + need < d[now.i - 1][now.j + 1][8])
        {
```

```
            temp.cost = now.cost + need;
            temp.i = now.i - 1;
            temp.j = now.j + 1;
            temp.flag = 8;
            q.push(temp);
            d[now.i - 1][now.j + 1][8] = now.cost + need;
}
                    }
                    //verh
                    need = abs(-1 - vi) + abs(vj);
                    if ((now.i - 1 >= 0))
                    {
if (now.cost + need < d[now.i - 1][now.j][0])
{
            temp.cost = now.cost + need;
            temp.i = now.i - 1;
            temp.j = now.j;
            temp.flag = 0;
            q.push(temp);
            d[now.i - 1][now.j][0] = now.cost + need;
}
if (now.cost + need < d[now.i - 1][now.j][1])
{
            temp.cost = now.cost + need;
            temp.i = now.i - 1;
            temp.j = now.j;
            temp.flag = 1;
            q.push(temp);
            d[now.i - 1][now.j][1] = now.cost + need;
}
                    }
            }
            else
            {
                    if (now.flag == 1 && now.i - 1 >= 0)
                    {
if (now.cost + 1 < d[now.i - 1][now.j][0])
{
            temp.cost = now.cost + 1;
            temp.i = now.i - 1;
            temp.j = now.j;
            temp.flag = 0;
            q.push(temp);
            d[now.i - 1][now.j][0] = now.cost + 1;
}
if (now.cost + 1 < d[now.i - 1][now.j][1])
{
            temp.cost = now.cost + 1;
            temp.i = now.i - 1;
            temp.j = now.j;
            temp.flag = 1;
            q.push(temp);
            d[now.i - 1][now.j][1] = now.cost + 1;
}
                    }
                    else
if (now.flag == 2 && (now.i - 1 >= 0) && (now.j - 1 >= 0))
{
            if (now.cost + 2 < d[now.i - 1][now.j - 1][0])
            {
                    temp.cost = now.cost + 2;
                    temp.i = now.i - 1;
                    temp.j = now.j - 1;
                    temp.flag = 0;
                    q.push(temp);
                    d[now.i - 1][now.j - 1][0] = now.cost + 2;
            }
            if (now.cost + 2 < d[now.i - 1][now.j - 1][2])
            {
                    temp.cost = now.cost + 2;
                    temp.i = now.i - 1;
                    temp.j = now.j - 1;
                    temp.flag = 2;
                    q.push(temp);
```

```
                d[now.i - 1][now.j - 1][2] = now.cost + 2;
        }
}
else
        if (now.flag == 3 && now.j - 1 >= 0)
        {
                if (now.cost + 1 < d[now.i][now.j - 1][0])
                {
                        temp.cost = now.cost + 1;
                        temp.i = now.i;
                        temp.j = now.j - 1;
                        temp.flag = 0;
                        q.push(temp);
                        d[now.i][now.j - 1][0] = now.cost + 1;
                }
                if (now.cost + 1 < d[now.i][now.j - 1][3])
                {
                        temp.cost = now.cost + 1;
                        temp.i = now.i;
                        temp.j = now.j - 1;
                        temp.flag = 3;
                        q.push(temp);
                        d[now.i][now.j - 1][3] = now.cost + 1;
                }
        }
        else
                if (now.flag == 4 && (now.i + 1 < n) && (now.j - 1 >= 0))
                {
                        if (now.cost + 2 < d[now.i + 1][now.j - 1][0])
                        {
        temp.cost = now.cost + 2;
        temp.i = now.i + 1;
        temp.j = now.j - 1;
        temp.flag = 0;
        q.push(temp);
        d[now.i + 1][now.j - 1][0] = now.cost + 2;
                        }
                        if (now.cost + 2 < d[now.i + 1][now.j - 1][4])
                        {
        temp.cost = now.cost + 2;
        temp.i = now.i + 1;
        temp.j = now.j - 1;
        temp.flag = 4;
        q.push(temp);
        d[now.i + 1][now.j - 1][4] = now.cost + 2;
                        }
                }
                else
                        if (now.flag == 5 && now.i + 1 < n)
                        {
        if (now.cost + 1 < d[now.i + 1][now.j][0])
        {
                temp.cost = now.cost + 1;
                temp.i = now.i + 1;
                temp.j = now.j;
                temp.flag = 0;
                q.push(temp);
                d[now.i + 1][now.j][0] = now.cost + 1;
        }
        if (now.cost + 1 < d[now.i + 1][now.j][5])
        {
                temp.cost = now.cost + 1;
                temp.i = now.i + 1;
                temp.j = now.j;
                temp.flag = 5;
                q.push(temp);
                d[now.i + 1][now.j][5] = now.cost + 1;
        }
                        }
                        else
        if (now.flag == 6 && (now.i + 1 < n) && (now.j + 1 < m))
        {
                if (now.cost + 2 < d[now.i + 1][now.j + 1][0])
                {
```

```
                                temp.cost = now.cost + 2;
                                temp.i = now.i + 1;
                                temp.j = now.j + 1;
                                temp.flag = 0;
                                q.push(temp);
                                d[now.i + 1][now.j + 1][0] = now.cost + 2;
                        }
                        if (now.cost + 2 < d[now.i + 1][now.j + 1][6])
                        {
                                temp.cost = now.cost + 2;
                                temp.i = now.i + 1;
                                temp.j = now.j + 1;
                                temp.flag = 6;
                                q.push(temp);
                                d[now.i + 1][now.j + 1][6] = now.cost + 2;
                        }
                }
                else
                        if (now.flag == 7 && now.j + 1 < m)
                        {
                                if (now.cost + 1 < d[now.i][now.j + 1][0])
                                {
                                        temp.cost = now.cost + 1;
                                        temp.i = now.i;
                                        temp.j = now.j + 1;
                                        temp.flag = 0;
                                        q.push(temp);
                                        d[now.i][now.j + 1][0] = now.cost + 1;
                                }
                                if (now.cost + 1 < d[now.i][now.j + 1][7])
                                {
                                        temp.cost = now.cost + 1;
                                        temp.i = now.i;
                                        temp.j = now.j + 1;
                                        temp.flag = 7;
                                        q.push(temp);
                                        d[now.i][now.j + 1][7] = now.cost + 1;
                                }
                        }
                        else
                                if (now.flag == 8 && (now.i - 1 >= 0) && (now.j + 1 < m))
                                {
                                        if (now.cost + 2 < d[now.i - 1][now.j + 1][0])
                                        {
                        temp.cost = now.cost + 2;
                        temp.i = now.i - 1;
                        temp.j = now.j + 1;
                        temp.flag = 0;
                        q.push(temp);
                        d[now.i - 1][now.j + 1][0] = now.cost + 2;
                                        }
                                        if (now.cost + 2 < d[now.i - 1][now.j + 1][8])
                                        {
                        temp.cost = now.cost + 2;
                        temp.i = now.i - 1;
                        temp.j = now.j + 1;
                        temp.flag = 8;
                        q.push(temp);
                        d[now.i - 1][now.j + 1][8] = now.cost + 2;
                                        }
                                }
                }
        }
}
```

Task E ()

```
#include <iostream>
#include <queue>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <cmath>
#include <algorithm>
#include <string>
#include <stack>
#define int long long

using namespace std;

signed main()
{
	int n, m, B;
	cin >> n >> m;
	pair<int, int> a[3];
	a[0] = make_pair(1, 1);
	a[1] = make_pair(1, m + 1);
	a[2] = make_pair(1, 2 * m + 1);
	a[3] = make_pair(1, 3 * m + 1);
	vector <pair<int, int>> need(3);
	for (int i = 0; i < B; i++)
		cin >> a[i].first >> a[i].second;
	int used[3] = { 0 };
	if (B == 1)
	{
		cout << "? " << a[0].first << " " << a[0].second << " " << a[0].first << " " << a
		    [0].second + m << endl;
		int a1, b1;
		cin >> a1 >> b1;
		if (b1 > m)
			cout << "! 1 1" << endl;
		else
			cout << "! 1 " << m + 1 << endl;
	}
	/*else
	{
		cout << "? " << a[0].first << " " << a[0].second << " " << a[0].first << " " << a
		    [0].second + m;
		int a1, b1;
		cin >> a1 >> b1;
		if (b1 > m)
			used[1] = 1;
		else
			used[0] = 1;
		cout << "? " << a[0].first << " " << a[0].second + 2 * m << " " << a[0].first << "
		     " << a[0].second + 3 * m;
		cin >> a1 >> b1;
		if (b1 <= m)
			used[0] = 1;
		else
			if (b1 <= 2 * m)
				used[1] = 1;
			else
				if (b1 <= 3 * m)
					used[2] = 1;
				else
					used[3] = 1;
		if(!used[0])

	}*/
}
```

Task F ()