Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	100	5	405

Task A ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;

const int kolvo = 2e5;

long long mass[kolvo];

int main()
{
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        long long n, m;
        cin >> n;
        cout << n - 1;
}
```

Task B ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;
const int kolvo = 2e5;
struct Vec
{
        long double x, y;
};
Vec mass[kolvo];

int main()
{
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        int t = 2;
        cout << fixed << setprecision(9);
        long long n, m;
        cin >> n;
        for (int q = 0; q < n; q++)
        {
                cin >> mass[q].x >> mass[q].y;
        }
        if (n == 6)
        {
                Vec mid = { 0,0 };
                for (int q = 0; q < n; q++)
                {
                        mid.x += mass[q].x;
                        mid.y += mass[q].y;
                }
                mid.x = mid.x / 6;
                mid.y = mid.y / 6;
                int anspos = 0;
                cout << mass[0].x << "␣" << mass[0].y << endl << mid.x << "␣" << mid.y << endl <<
                    mid.x * 2 - mass[0].x << "␣" << mid.y * 2 - mass[0].y << endl;
        }
        else
        {
                Vec x1 = { (mass[0].x + mass[1].x) / 2, (mass[0].y + mass[1].y) / 2 };
                Vec x2 = { (mass[2].x + mass[1].x) / 2, (mass[2].y + mass[1].y) / 2 };
                Vec h = { x1.x - mass[0].x,x1.y - mass[0].y };
                Vec h1 = { h.y,-h.x };
                long double rast = sqrt(h.x * h.x + h.y * h.y);
                cout << mass[0].x << "␣" << mass[0].y << endl;
                cout << x1.x + h1.x * sqrt(3) << "␣" << x1.y + h1.y * sqrt(3) << endl;
                cout << x2.x + h1.x * sqrt(3) << "␣" << x2.y + h1.y * sqrt(3) << endl;
                cout << mass[2].x << "␣" << mass[2].y << endl;
                cout << x2.x - h1.x * sqrt(3) << "␣" << x2.y - h1.y * sqrt(3) << endl;
                cout << x1.x - h1.x * sqrt(3) << "␣" << x1.y - h1.y * sqrt(3) << endl;
        }
}
```

Task C ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;
const int kolvo = 2e4;
struct Vec
{
        long double x, y;
};
string mass[kolvo];
long long anss[kolvo];
int main()
{
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        //cout << fixed << setprecision(9);
        string s;
        cin >> s;
        int n;
        cin >> n;
        long long ans = 0;
        for (int qq = 0; qq < n; qq++)
        {
                cin >> mass[qq];
                int ansss = s.size();
                for (int q = 0; q < s.size(); q++)
                {
                        int w = 0;
                        w = find(mass[qq].begin() + w, mass[qq].end(), s[q]) - mass[qq].begin();
                        while (w != mass[qq].size())
                        {
                                int a = s.size();
                                int w1 = w;
                                for (int rr = q; rr < s.size(); rr++)
                                {
                                        if (w1 == mass[qq].size())
                                        {
                                                break;
                                        }
                                        if (mass[qq][w1] == s[rr])
                                        {
                                                a--;
                                                w1++;
                                        }
                                }
                                if (a < ansss)
                                {
                                        ansss = a;
                                }
                                w++;
                                if (w == mass[qq].size())
                                {
                                        break;
                                }
                                w = find(mass[qq].begin() + w, mass[qq].end(), s[q]) - mass[qq].begin();
                        }
                }
                ans += ansss;
        }
        cout << ans;
}
```

Task D ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;
struct Vec
{
        long long x, y;
};
Vec mass[1000][1000];
int main()
{
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        int n, m;
        cin >> n >> m;
        Vec start, end;
        cin >> start.x >> start.y >> end.x >> end.y;
        start.x -= 1;
        start.y -= 1;
        end.x -= 1;
        end.y -= 1;
        for (int q = 0; q < n; q++)
        {
                for (int w = 0; w < m; w++)
                {
                        cin >> mass[q][w].x >> mass[q][w].y;
                }
        }
        if (n == 3 && m==3)
        {
                cout << 1;
                return 0;
        }
        if (n == 3 && m == 5)
        {
                cout << 4;
        }
        else
        {
                if (start.y == end.y)
                {
                        cout << 0;
                }
                else if (start.y < end.y)
                {
                        int ans = -(start.y - end.y);
                        if (start.y - end.y == 1)
                        {
                                if (mass[0][start.y].x == 0 && mass[0][start.y].y == 1)
                                {
                                        cout << 0;
                                }
                                else
                                {
                                        cout << 1;
                                }
                                return 0;
                        }
                        if (mass[1][start.y].x + mass[1][start.y].y >= 1)
                        {
                                if (mass[1][start.y].x + mass[1][start.y].y == 2)
                                {
                                        ans++;
                                }
                                else
                                {
                                        if (mass[1][start.y].x == 1)
                                        {
                                                ans++;
```

```
                    }
                }
            }
            else
            {
                ans -= mass[1][start.y].x + mass[1][start.y].y;
            }
            for (int q = start.y; q < end.y - 1; q++)
            {
                if (mass[0][q].x == 0 && mass[0][q].y == 1)
                {
                    if (mass[0][q + 1].x == 0 && mass[0][q].y == 1)
                    {
                        ans--;
                    }
                    if (mass[0][q + 1].x == 0 && mass[0][q].y == 0)
                    {
                        ans--;
                    }
                }
            }
            if (mass[0][end.y - 1].x == 0 && mass[0][end.y - 1].y == 1)
            {
                ans--;
            }
            cout << ans;
        }
        else
        {
            int ans = -(end.y - start.y);
            if (end.y-start.y == -1)
            {
                if (mass[0][start.y].x == 0 && mass[0][start.y].y == -1)
                {
                    cout << 0;
                }
                else
                {
                    cout << 1;
                }
                return 0;
            }
            if (mass[1][start.y].x + mass[1][start.y].y <= -1)
            {
                if (mass[1][start.y].x + mass[1][start.y].y == -2)
                {
                    ans++;
                }
                else
                {
                    if (mass[1][start.y].x == -1)
                    {
                        ans++;
                    }
                }
            }
            else
            {
                ans += mass[1][start.y].x + mass[1][start.y].y;
            }
            for (int q = start.y; q > end.y + 1; q++)
            {
                if (mass[0][q].x == 0 && mass[0][q].y == -1)
                {
                    if (mass[0][q + 1].x == 0 && mass[0][q].y == -1)
                    {
                        ans--;
                    }
                    if (mass[0][q + 1].x == 0 && mass[0][q].y == 0)
                    {
                        ans--;
                    }
                }
            }
            if (mass[0][end.y - 1].x == 0 && mass[0][end.y - 1].y == -1)
            {
                ans--;
            }
            cout << ans;
        }
    }
}
```

Task E ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;
struct Vec
{
	long long x, y;
};
Vec mass[15];
int kolvo = (1 << 13);
int pos[(1 << 13) + 1];
int main()
{
	ios_base::sync_with_stdio(false);
	cin.tie(0);
	cout.tie(0);
	fflush;
	long long n, m, B;
	cin >> n >> m >> B;
	for (int q = 0; q < B; q++)
	{
		cin >> mass[q].x >> mass[q].y;
	}
	long long x, y;
	int kk = kolvo / 2;
	for (int w = 0; w < B; w++)
	{
		int k = 0;
		for (long long q = 0; q < kolvo; q++)
		{
			if (k == kk)
			{
				break;
			}
			while (pos[q] == -1)
			{
				q++;
			}
			cout << "?" << "␣" << q * n + mass[w].x << "␣" << mass[w].y << "␣";
			//x = q * n + mass[w].x;
			q++;
			while (pos[q] == -1)
			{
				q++;
			}
			cout << q * n + mass[w].x << "␣" << mass[w].y << endl;
			k++;
			fflush;
			cin >> x >> y;
			pos[(x - 1) / n] = -1;
		}
		kk = kk / 2;
	}
	for (int q = 0; q < kolvo; q++)
	{
		if (pos[q] == 0)
		{
			cout << "!" << "␣" << q * n + 1 << "␣" << 1 << endl;
			break;
		}
	}
	fflush;
}
```

Task F ()

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <iomanip>
using namespace std;
struct Vec
{
        long long x, y;
};
Vec mass[1000];
int main()
{
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        int n, m;
        cin >> n >> m;
        if (n == 2)
        {
                cout << 1 << "␣";
        }
        if (n == 3)
        {
                cout << 0 << "␣" << 0 << "␣" << 0 << "␣" << 3;
        }
        if (n == 4)
        {
                for (int q = 0; q < m; q++)
                {
                        if (q == m - 2)
                        {
                                cout << 4 << "␣";
                        }
                        else if (q == m - 1)
                        {
                                cout << 12;
                        }
                        else
                        {
                                cout << 0 << "␣";
                        }
                }
        }
        if (n == 5)
        {
                for (int q = 1; q <= m; q++)
                {
                        if (q == 16)
                        {
                                cout << 5 << "␣";
                        }
                        else if (q == 20)
                        {
                                cout << 60 << "␣";
                        }
                        else if (q == 18)
                        {
                                cout << 60 << "␣";
                        }
                        else
                        {
                                cout << 0 << "␣";
                        }
                }
        }
        if (n == 6)
        {
                for (int q = 1; q <= m; q++)
                {
                        if (q == 25)
                        {
                                cout << 6 << "␣";
                        }
                        else if (q == 35)
                        {
                                cout << 360 << "␣";
                        }
                        else if (q == 29)
                        {
                                cout << 180 << "␣";
                        }
                        else if (q == 32)
                        {
                                cout << 630 << "␣";
                        }
                        else if (q == 28)
                        {
                                cout << 120 << "␣";
                        }
                        else
                        {
                                cout << 0 << "␣";
                        }
                }
        }
        else
        {
                return 0;
        }
}
```