Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	60	100	100	25	485

Task A ()

```
#include<iostream>
#include<algorithm>
#include<vector>
#include<string>

using namespace std;

int main()
{
        int k;
        cin >> k;
        if (k == 1)
        {
                cout << 1;
                return 0;
    }
        cout << ((k - 2) % 9 + 2) % 10;
}
```

Task B ()

```cpp
#include<iostream>
#include<algorithm>
#include<vector>
#include<string>
#include<unordered_set>

using namespace std;

int main()
{
	ios_base::sync_with_stdio(false);
	cin.tie(0);
	cout.tie(0);

	int n, k;
	cin >> n >> k;
	string s;
	cin >> s;
	unordered_set<char> c;
	int len = 0, ans = 1;
	for (char i : s)
	{
		c.insert(i);
		++len;
		if ((int)c.size() > 3 || len > k)
		{
			++ans;
			c.clear();
			c.insert(i);
			len = 1;
		}
	}
	cout << ans;
}
```

Task C ()

```
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
#pragma GCC target("avx2")

#include<iostream>
#include<algorithm>
#include<vector>
#include<string>

using namespace std;

const int INF = 1e9;

int main()
{
	ios_base::sync_with_stdio(false);
	cin.tie(0);
	cout.tie(0);

	int n, x, y;
	cin >> n >> x >> y;
	vector<int> v(n), w(n);
	int summ = 0;
	for (int i = 0; i < n; ++i)
	{
		cin >> v[i];
	}
	for (int i = 0; i < n; ++i)
	{
		cin >> w[i];
		summ += w[i];
	}
	vector<vector<int>> dp(n, vector<int>(x + 1, -INF));
	vector<vector<string>> s(2, vector<string>(x + 1, ""));
	dp[0][0] = 0;
	s[0][0] = "y";
	if (v[0] <= x)
	{
		dp[0][v[0]] = w[0];
		s[0][v[0]] = "x";
	}
	for (int i = 1; i < n; ++i)
	{
		for (int j = 0; j < min(v[i], x + 1); ++j)
		{
			dp[i % 2][j] = dp[(i - 1) % 2][j];
			s[i % 2][j] = s[(i - 1) % 2][j] + 'y';
		}
		for (int j = v[i]; j < x + 1; ++j)
		{
			dp[i % 2][j] = dp[(i - 1) % 2][j];
			s[i % 2][j] = s[(i - 1) % 2][j] + 'y';
			if (dp[(i - 1) % 2][j - v[i]] + w[i] >= dp[i % 2][j])
			{
				dp[i % 2][j] = dp[(i - 1) % 2][j - v[i]] + w[i];
				s[i % 2][j] = s[(i - 1) % 2][j - v[i]] + 'x';
			}
		}
	}
	int pos = -1;
	for (int i = 0; i < x + 1; ++i)
	{
		if (dp[(n - 1) % 2][i] != -INF && pos == -1 || dp[(n - 1) % 2][i] >= dp[(n - 1) %
		    2][pos])
		{
			pos = i;
		}
	}
	if (pos == -1 || summ - dp[(n - 1) % 2][pos] > y)
	{
		cout << -1;
		return 0;
```

```
        }
        cout << s[(n - 1) % 2][pos];
}
```

Task D ()

```
#include<iostream>
#include<algorithm>
#include<vector>

using namespace std;

int solve(vector<int> v)
{
	int n = (int)v.size();
	if (n == 0)
	{
		return 0;
	}
	vector<pair<int, int>> loc;
	int cnt = 1;
	for (int i = 1; i < n; ++i)
	{
		if (v[i] == v[i - 1])
		{
			++cnt;
		}
		else
		{
			loc.push_back({ cnt % 2, v[i - 1] });
			cnt = 1;
		}
	}
	loc.push_back({ cnt % 2, v[n - 1] });
	vector<int> loc2;
	for (auto i : loc)
	{
		if (i.first)
		{
			loc2.emplace_back(i.second);
		}
	}
	if ((int)loc.size() == (int)loc2.size())
	{
		return (int)loc.size() / 2;
	}
	return solve(loc2);
}

int main()
{
	int n;
	cin >> n;
	vector<int> a(2 * n);
	for (int i = 0; i < 2 * n; ++i)
	{
		char x;
		cin >> x;
		if (x == '(' || x == ')')
		{
			a[i] = 0;
		}
		else
		{
			a[i] = 1;
		}
	}
	cout << solve(a);
}
```

Task E ()

```
#include<iostream>
#include<algorithm>
#include<vector>

using namespace std;

int main()
{
	ios_base::sync_with_stdio(false);
	cin.tie(0);
	cout.tie(0);

	string type;
	cin >> type;
	if (type == "add")
	{
		int t;
		cin >> t;
		while (t--)
		{
			int n, k;
			cin >> n >> k;
			vector<bool> u(n + 1);
			int summ = 0;
			for (int i = 0; i < k; ++i)
			{
				int x;
				cin >> x;
				u[x] = true;
				summ += x;
				summ %= (k + 1);
			}
			int cnt = 0;
			bool tf = true;
			for (int i = 1; i < n + 1 && tf; ++i)
			{
				if (u[i])
				{
					++cnt;
					continue;
				}
				if ((summ + i) % (k + 1) == cnt)
				{
					cout << i << '\n';
					tf = false;
				}
			}
		}
	}
	else
	{
		int t;
		cin >> t;
		while (t--)
		{
			int n, k;
			cin >> n >> k;
			vector<int> loc(k + 1);
			int summ = 0;
			for (int i = 0; i < k + 1; ++i)
			{
				cin >> loc[i];
				summ += loc[i];
				summ %= (k + 1);
			}
			sort(loc.begin(), loc.end());
			for (int i = 0; i < summ; ++i)
			{
				cout << loc[i] << ' ';
			}
			for (int i = summ + 1; i < k + 1; ++i)
			{
```

```
                        cout << loc[i] << ' ';
                    }
                    cout << '\n';
                }
            }
        }
```

Task F ()

```cpp
#include<iostream>
#include<algorithm>
#include<vector>

using namespace std;

int main()
{
	int n;
	cin >> n;
	int m = 4;
	vector<vector<int>> p = { {0, 0}, {1, 0}, {1, 1}, {0, 1} },
		v = { {1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0}, {-1, -1}, {0, -1}, {1, -1} };
	cout << m << '\n';
	for (auto i : p)
	{
		cout << i[0] << ' ' << i[1] << '\n';
	}
	for (int i = 0; i < n; ++i)
	{
		cout << v[i][0] << ' ' << v[i][1] << '\n';
	}
}
```