Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	20	25	445

Task A ()

```
x = int(input())-1

if x == 0:
    print('1')
else:
    x -= 1
    print('234567890'[x % 9])
```

Task B ()

```
#include <iostream>

using namespace std;

int main() {
    int n, k;
    cin >> n >> k;

    char x;
    int len = 0;
    int ans = 1;
    char first = '\0';
    char second = '\0';
    char third = '\0';
    for (int i = 0; i < n; ++i) {
        if(len == k){
            len = 0;
            ans++;
            first = '\0';
            second = '\0';
            third = '\0';
        }
        cin >> x;
        len++;

        if(first != x){
            if(first == '\0'){
                first = x;
                continue;
            }
            if(second != x){
                if(second == '\0'){
                    second = x;
                    continue;
                }
                if(third != x){
                    if(third == '\0'){
                        third = x;
                        continue;
                    }
                    ans++;
                    len = 1;
                    first = x;
                    second = '\0';
                    third = '\0';
                }
            }
        }

    }

    cout << ans;

    return 0;
}
```

Task C ()

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

using namespace std;

int main(){
    int n, x, y;
    cin >> n >> x >> y;

    int total_size = 0, total_weight = 0;
    vector<pair<int, int>> items(n); // V, W
    for (int i = 0; i < n; ++i) {
        cin >> items[i].first;
        total_size += items[i].first;
    }
    for (int i = 0; i < n; ++i) {
        cin >> items[i].second;
        total_weight += items[i].second;
    }

    vector<vector<int>> knapsack(2, vector<int>(y + 1, -1));
    vector<vector<int>> pr(n+1, vector<int>(y + 1, -1));
    knapsack[0][0] = 0;

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j <= y; ++j) {
            if(knapsack[0][j] > -1){
                if(knapsack[1][j] < knapsack[0][j]){
                    knapsack[1][j] = knapsack[0][j];
                    pr[i+1][j] = j;
                }
                if(j + items[i].second <= y){
                    if(knapsack[1][j + items[i].second] < knapsack[0][j] + items[i].first) {
                        knapsack[1][j + items[i].second] = knapsack[0][j] + items[i].first;
                        pr[i+1][j + items[i].second] = j;
                    }
                }
            }
        }
        swap(knapsack[0], knapsack[1]);
        knapsack[1].assign(y + 1, -1);
    }

    for (int i = 0; i <= y; ++i) {
        if(knapsack[0][i] > -1){
            if(total_size - knapsack[0][i] <= x){
                int l = n;
                int m = i;
                vector<bool> ans(n, false);
                while(l > 0){
                    if(pr[l][m] < m){
                        ans[l-1] = true;
                        m = pr[l][m];
                    }
                    l--;
                }

                for(bool j: ans){
                    if(j){
                        cout << 'y';
                    }
                    else{
                        cout << 'x';
                    }
                }
                cout << '\n';

                return 0;
            }
        }
    }
    cout << -1 << '\n';
    return 0;
}
```

Task D ()

```cpp
#include <iostream>
#include <stack>

using namespace std;

int main(){
    int n;
    cin >> n;
    n *= 2;

    stack<bool> stak;
    int ans = 0;

    char x;
    for (int i = 0; i < n; ++i) {
        cin >> x;
        switch (x) {
            case '(':
                if(stak.empty() || stak.top()){
                    stak.push(false);
                }
                else{
                    stak.pop();
                }
                break;
            case '[':
                if(stak.empty() || !stak.top()){
                    stak.push(true);
                }
                else{
                    stak.pop();
                }
                break;
            case ')':
                if(stak.empty() || stak.top()){
                    stak.push(false);
                }
                else{
                    stak.pop();
                }
                break;
            case ']':
                if(stak.empty() || !stak.top()){
                    stak.push(true);
                }
                else{
                    stak.pop();
                }
                break;
            default:
                i--;
                break;
        }
    }

    cout << stak.size() / 2;


    return 0;
}
```

Task E ()

```
import random

typ = input().strip()
t = int(input())

dd =7
random.seed(12445657433)
kkk = random.randint(1, 100000)
for _ in range(t):
    n, k = map(int, input().strip().split())
    numbers = set(map(int, input().strip().split()))
    if typ == 'add':
        if k > 10:
            print(kkk)
            continue

        random.seed(sum(numbers) * dd)
        x = random.randint(1, n)
        while x in numbers:
            x = random.randint(1, n)
        print(x)
    else:
        if k > 10:
            numbers.remove(kkk)
            for j in numbers:
                print(j, end='␣')
            print()
            continue

        summ = sum(numbers) * dd
        for i in numbers:
            random.seed(summ - (i * dd))
            x = random.randint(1, n)
            while x != i and x in numbers:
                x = random.randint(1, n)
            if x == i:
                numbers.remove(i)
                for j in numbers:
                    print(j, end='␣')
                print()
                break
```

Task F ()

```
n = int(input())
print(4)
print(0, 0)
print(1, 0)
print(1, 1)
print(0, 1)

ans = [
    (1, 0),
    (1, 1),
    (0, 1),
    (-1, 1),
    (-1, 0),
    (-1, -1),
    (0, -1),
    (1, -1)
]

for i in ans[:n]:
    print(*i)
```