Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	25	480

Task A ()

```python
a = int(input())

if (a <= 9):
        print(a)
else:
        a -= 10
        a = a % 9

        if (a != 0):
                a += 1

        print(a)
```

Task B ()

```cpp
#include<stdio.h>
#include<iostream>
#include<set>
#include<vector>
#include<map>
#include<string>
#include<algorithm>
#define ll long long
#define mp make_pair
using namespace std;
int dp[100001];
map<char, int> m;
int main()
{
        int k;
        string s;
        cin >> k >> k >> s;
        int maxLen = 0, sLen = 0;

        for (int i = 1; i <= s.size(); i++)
                dp[i] = 1e9;

        for (int i = 0; i < s.size(); i++)
        {
                while (maxLen < s.size() && sLen < k)
                {
                        if (m.find(s[maxLen]) == m.end())
                        {
                                if (m.size() == 3)
                                        break;

                                m[s[maxLen]] = 1;
                        }
                        else
                                m[s[maxLen]]++;

                        dp[maxLen + 1] = min(dp[i] + 1, dp[maxLen + 1]);
                        maxLen++;
                        sLen++;
                }

                m[s[i]]--;

                if (m[s[i]] == 0)
                        m.erase(m.find(s[i]));

                sLen--;
        }

        cout << dp[s.size()];
}
```

Task C ()

```cpp
#include<stdio.h>
#include<iostream>
#include<set>
#include<vector>
#include<map>
#include<string>
#include<algorithm>
#define ll long long
#define mp make_pair
using namespace std;
int dp[2][250000], v[500], w[500];
bool his[500][250001];
int main()
{
        int n, x, y;
        cin >> n >> x >> y;

        for (int i = 0; i < n; i++)
                cin >> v[i];

        for (int i = 0; i < n; i++)
                cin >> w[i];

        for (int i = 0; i < 2; i++)
                for (int j = 0; j < 250000; j++)
                        dp[i][j] = 1e9;

        dp[0][0] = 0;

        for (int i = 0; i < n; i++)
        {
                int dpInd = i % 2, nextDp = (i + 1) % 2;

                for (int j = 0; j <= x; j++)
                {
                        if (dp[dpInd][j] < 1e9)
                        {
                                if (j + v[i] <= x)
                                {
                                        if (dp[nextDp][j + v[i]] > dp[dpInd][j])
                                        {
                                                dp[nextDp][j + v[i]] = dp[dpInd][j];
                                                his[i][j + v[i]] = true;
                                        }
                                }

                                if (dp[nextDp][j] > dp[dpInd][j] + w[i])
                                {
                                        dp[nextDp][j] = dp[dpInd][j] + w[i];
                                        his[i][j] = false;
                                }
                        }

                        dp[dpInd][j] = 1e9;
                }
        }

        int ansInd = 0;

        for (int i = 1; i <= x; i++)
                if (dp[n % 2][i] < dp[n % 2][ansInd])
                        ansInd = i;

        if (dp[n % 2][ansInd] > y)
        {
                cout << -1;
                return 0;
        }

        string ans;

        while (n != 0)
```

```cpp
	{
		n--;

		if (his[n][ansInd])
		{
			ans.push_back('x');
			ansInd -= v[n];
		}
		else
		{
			ans.push_back('y');
		}
	}

	reverse(ans.begin(), ans.end());

	cout << ans;
}
```

Task D ()

```cpp
#include<stdio.h>
#include<iostream>
#include<set>
#include<vector>
#include<map>
#include<string>
#include<algorithm>
#define ll long long
#define mp make_pair
using namespace std;
int main()
{
        int o = 0, e = 0;
        string s;

        cin >> s >> s;

        for (int i = 0; i < s.size(); i++)
        {
                if (s[i] == ']' || s[i] == '[')
                {
                        if (i % 2 == 0)
                                e++;

                        if (i % 2 == 1)
                                o++;
                }
        }

        cout << abs(e - o);
}
```

Task E ()

```python
ansAdd = {"123": 4, "124": 5, "125": 3, "126": 3, "127": 3, "128": 3, "129": 3, "1210": 3, "134":
    5, "135": 6, "136": 4, "137": 4, "138": 4, "139": 4, "1310": 4, "145": 6, "146": 2, "147": 2,
    "148": 2, "149": 2, "1410": 2, "156": 2, "157": 2, "158": 2, "159": 2, "1510": 2, "167": 2, "
    168": 2, "169": 2, "1610": 2, "178": 2, "179": 2, "1710": 2, "189": 2, "1810": 2, "1910": 2, "
    234": 5, "235": 6, "236": 4, "237": 4, "238": 4, "239": 4, "2310": 4, "245": 6, "246": 7, "247
    ": 5, "248": 5, "249": 5, "2410": 5, "256": 7, "257": 3, "258": 3, "259": 3, "2510": 3, "267":
    3, "268": 3, "269": 3, "2610": 3, "278": 3, "279": 3, "2710": 3, "289": 3, "2810": 3, "2910":
    3, "345": 6, "346": 7, "347": 5, "348": 5, "349": 5, "3410": 5, "356": 7, "357": 1, "358": 1,
    "359": 1, "3510": 1, "367": 1, "368": 1, "369": 1, "3610": 1, "378": 1, "379": 1, "3710": 1,
    "389": 1, "3810": 1, "3910": 1, "456": 7, "457": 1, "458": 1, "459": 1, "4510": 1, "467": 1, "
    468": 1, "469": 1, "4610": 1, "478": 1, "479": 1, "4710": 1, "489": 1, "4810": 1, "4910": 1, "
    567": 1, "568": 1, "569": 1, "5610": 1, "578": 1, "579": 1, "5710": 1, "589": 1, "5810": 1, "
    5910": 1, "678": 1, "679": 1, "6710": 1, "689": 1, "6810": 1, "6910": 1, "789": 1, "7810": 1,
    "7910": 1, "8910": 1}
ansClear = {"1234": "1_2_3", "1245": "1_2_4", "1235": "1_2_5", "1236": "1_2_6", "1237": "1_2_7", "
    1238": "1_2_8", "1239": "1_2_9", "12310": "1_2_10", "1345": "1_3_4", "1356": "1_3_5", "1346":
    "1_3_6", "1347": "1_3_7", "1348": "1_3_8", "1349": "1_3_9", "13410": "1_3_10", "1456": "1_4_5"
    , "1246": "1_4_6", "1247": "1_4_7", "1248": "1_4_8", "1249": "1_4_9", "12410": "1_4_10", "1256
    ": "1_5_6", "1257": "1_5_7", "1258": "1_5_8", "1259": "1_5_9", "12510": "1_5_10", "1267": "1_6
    _7", "1268": "1_6_8", "1269": "1_6_9", "12610": "1_6_10", "1278": "1_7_8", "1279": "1_7_9", "
    12710": "1_7_10", "1289": "1_8_9", "12810": "1_8_10", "12910": "1_9_10", "2345": "2_3_4", "
    2356": "2_3_5", "2346": "2_3_6", "2347": "2_3_7", "2348": "2_3_8", "2349": "2_3_9", "23410": "
    2_3_10", "2456": "2_4_5", "2467": "2_4_6", "2457": "2_4_7", "2458": "2_4_8", "2459": "2_4_9",
    "24510": "2_4_10", "2567": "2_5_6", "2357": "2_5_7", "2358": "2_5_8", "2359": "2_5_9", "23510"
    : "2_5_10", "2367": "2_6_7", "2368": "2_6_8", "2369": "2_6_9", "23610": "2_6_10", "2378": "2_7
    _8", "2379": "2_7_9", "23710": "2_7_10", "2389": "2_8_9", "23810": "2_8_10", "23910": "2_9_10"
    , "3456": "3_4_5", "3467": "3_4_6", "3457": "3_4_7", "3458": "3_4_8", "3459": "3_4_9", "34510"
    : "3_4_10", "3567": "3_5_6", "1357": "3_5_7", "1358": "3_5_8", "1359": "3_5_9", "13510": "3_5_
    10", "1367": "3_6_7", "1368": "3_6_8", "1369": "3_6_9", "13610": "3_6_10", "1378": "3_7_8", "
    1379": "3_7_9", "13710": "3_7_10", "1389": "3_8_9", "13810": "3_8_10", "13910": "3_9_10", "
    4567": "4_5_6", "1457": "4_5_7", "1458": "4_5_8", "1459": "4_5_9", "14510": "4_5_10", "1467":
    "4_6_7", "1468": "4_6_8", "1469": "4_6_9", "14610": "4_6_10", "1478": "4_7_8", "1479": "4_7_9"
    , "14710": "4_7_10", "1489": "4_8_9", "14810": "4_8_10", "14910": "4_9_10", "1567": "5_6_7", "
    1568": "5_6_8", "1569": "5_6_9", "15610": "5_6_10", "1578": "5_7_8", "1579": "5_7_9", "15710":
    "5_7_10", "1589": "5_8_9", "15810": "5_8_10", "15910": "5_9_10", "1678": "6_7_8", "1679": "6_
    7_9", "16710": "6_7_10", "1689": "6_8_9", "16810": "6_8_10", "16910": "6_9_10", "1789": "7_8_9
    ", "17810": "7_8_10", "17910": "7_9_10", "18910": "8_9_10"}
mode = input()
t = int(input())

for i in range(t):
        n, k = map(int, input().split())
        data = list(map(int, input().split()))

        if mode == 'add':
                if n == 10:
                        s = list(map(str, sorted(data)))
                        sAns = ""

                        for it in range(3):
                                sAns += s[it]

                        s = sAns

                        print(ansAdd[s])
                else:
                        dataSum = sum(data)
                        data = sorted(data)
                        lowerBound = 1
                        ans = []

                        for j in range(k):
                                for it in range(lowerBound, data[j]):
                                        if (dataSum + it) % 11 == j:
                                                ans.append(it)
                                lowerBound = data[j] + 1

                        for it in range(lowerBound, n + 1):
                                if (dataSum + it) % 11 == k:
                                        ans.append(it)

                        print(ans[0])
```

```python
if mode == 'clear':
        if n == 10:
                s = list(map(str, sorted(data)))
                sAns = ""

                for it in range(4):
                        sAns += s[it]

                s = sAns

                print(ansClear[s])
        else:
                dataSum = sum(data)
                data = sorted(data)

                for j in range(len(data)):
                        if j != dataSum % 11:
                                print(data[j], end=' ')

                print()
```

Task F ()

```python
vecs = ['-1_1', '-1_0', '-1_-1', '0_1', '0_-1', '1_-1', '1_0', '1_1']

n = int(input())

print('4\n1_1\n0_1\n0_0\n1_0')

for i in range(n):
        print(vecs[i])
```

```python
vecs = ['-1_1', '-1_0', '-1_-1', '0_1', '0_-1', '1_-1', '1_0', '1_1']

n = int(input())
```