

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	60	100	55	15	430

Task A ()

```
#include <bits/stdc++.h>

using namespace std;

#define int long long

int32_t main() {
    int k; cin >> k;
    if (k == 1) {
        cout << 1;
    }
    else {
        vector<int> ans = {2, 3, 4, 5, 6, 7, 8, 9, 0};
        cout << ans[(k - 2) % 9];
    }
}
```

Task B ()

```
n, k = map(int, input().split())
s = input()
i = 0
ans = 0
now = []
while i < n:
    j = i
    ok = True
    while j < n and j - i < k and ok:
        flag = True
        for c in now:
            if c == s[j]:
                flag = False
        if flag:
            now.append(s[j])
        if len(now) > 3:
            ans += 1
            now = []
            i = j
            ok = False
        j += 1
    if ok:
        now = []
        ans += 1
        i = j
print(ans)
```

Task C ()

```
#include <bits/stdc++.h>

using namespace std;

#define int long long

int32_t main() {
    int n, x, y; cin >> n >> x >> y;
    vector<int> v(n);
    for (int i = 0; i < n; i++) {
        cin >> v[i];
    }
    vector<int> w(n);
    for (int i = 0; i < n; i++) {
        cin >> w[i];
    }
    vector<vector<int>> dp(n + 1, vector<int>(x + 1, 0));
    for (int i = 1; i < n + 1; i++) {
        for (int j = 0; j < x + 1; j++) {
            dp[i][j] = dp[i - 1][j];
            if (j - v[i - 1] >= 0) {
                dp[i][j] = max(dp[i][j], dp[i - 1][j - v[i - 1]] + w[i - 1]);
            }
        }
    }
    int max_ = 0;
    int where = 0;
    for (int i = 0; i < x + 1; i++) {
        if (max_ < dp[n][i]) {
            max_ = dp[n][i];
            where = i;
        }
    }
    int sum_ = 0;
    for (int i = 0; i < n; i++) {
        sum_ += w[i];
    }
    if (sum_ - max_ > y) {
        cout << -1;
    }
    else {
        vector<int> ans;
        for (int i = n; i > 0; i--) {
            if (dp[i][where] == dp[i - 1][where]) {
                continue;
            }
            else {
                ans.push_back(i - 1);
                where -= v[i - 1];
            }
        }
        vector<bool> table(n, false);
        for (int i = 0; i < ans.size(); i++) {
            table[ans[i]] = true;
        }
        for (int i = 0; i < n; i++) {
            if (table[i]) {
                cout << 'x';
            }
            else {
                cout << 'y';
            }
        }
    }
}
```

Task D ()

```
n = int(input())
s1 = input()
ans = 0
s = []
for i in range(n * 2):
    if s1[i] == '(' or s1[i] == ')':
        s.append(0)
    else:
        s.append(1)
st = []
for i in range(n * 2):
    if len(st) == 0 or st[-1] != s[i]:
        st.append(s[i])
    else:
        st.pop()
print(len(st) // 2)
```

Task E ()

```
ansdict = {(1, 2, 3): 6, (1, 2, 3, 6): 6, (1, 2, 4): 5, (1, 2, 4, 5): 5, (1, 2, 5): 6, (1, 2, 5, 6): 6, (1, 2, 6): 10, (1, 2, 6, 10): 10, (1, 2, 7): 6, (1, 2, 6, 7): 6, (1, 2, 8): 6, (1, 2, 6, 8): 6, (1, 2, 9): 6, (1, 2, 6, 9): 6, (1, 2, 10): 5, (1, 2, 5, 10): 5, (1, 3, 4): 8, (1, 3, 4, 8): 8, (1, 3, 5): 2, (1, 2, 3, 5): 2, (1, 3, 6): 10, (1, 3, 6, 10): 10, (1, 3, 7): 2, (1, 2, 3, 7): 2, (1, 3, 8): 2, (1, 2, 3, 8): 2, (1, 3, 9): 2, (1, 2, 3, 9): 2, (1, 3, 10): 4, (1, 3, 4, 10): 4, (1, 4, 5): 8, (1, 4, 5, 8): 8, (1, 4, 6): 5, (1, 4, 5, 6): 5, (1, 4, 7): 8, (1, 4, 7, 8): 8, (1, 4, 8): 2, (1, 2, 4, 8): 2, (1, 4, 9): 8, (1, 4, 8, 9): 8, (1, 4, 10): 8, (1, 4, 8, 10): 8, (1, 5, 6): 10, (1, 5, 6, 10): 10, (1, 5, 7): 2, (1, 2, 5, 7): 2, (1, 5, 8): 2, (1, 2, 5, 8): 2, (1, 5, 9): 2, (1, 2, 5, 9): 2, (1, 5, 10): 4, (1, 4, 5, 10): 4, (1, 6, 7): 10, (1, 6, 7, 10): 10, (1, 6, 8): 5, (1, 5, 6, 8): 5, (1, 6, 9): 10, (1, 6, 9, 10): 10, (1, 6, 10): 4, (1, 4, 6, 10): 4, (1, 7, 8): 2, (1, 2, 7, 8): 2, (1, 7, 9): 2, (1, 2, 7, 9): 2, (1, 7, 10): 4, (1, 4, 7, 10): 4, (1, 8, 9): 2, (1, 2, 8, 9): 2, (1, 8, 10): 5, (1, 5, 8, 10): 5, (1, 9, 10): 4, (1, 4, 9, 10): 4, (2, 3, 4): 7, (2, 3, 4, 7): 7, (2, 3, 5): 6, (2, 3, 5, 6): 6, (2, 3, 6): 10, (2, 3, 6, 10): 10, (2, 3, 7): 6, (2, 3, 6, 7): 6, (2, 3, 8): 6, (2, 3, 6, 8): 6, (2, 3, 9): 6, (2, 3, 6, 9): 6, (2, 3, 10): 7, (2, 3, 7, 10): 7, (2, 4, 5): 9, (2, 4, 5, 9): 9, (2, 4, 6): 1, (1, 2, 4, 6): 1, (2, 4, 7): 1, (1, 2, 4, 7): 1, (2, 4, 8): 10, (2, 4, 8, 10): 10, (2, 4, 9): 3, (2, 3, 4, 9): 3, (2, 4, 10): 1, (1, 2, 4, 10): 1, (2, 5, 6): 10, (2, 5, 6, 10): 10, (2, 5, 7): 6, (2, 5, 6, 7): 6, (2, 5, 8): 6, (2, 5, 6, 8): 6, (2, 5, 9): 6, (2, 5, 6, 9): 6, (2, 5, 10): 9, (2, 5, 9, 10): 9, (2, 6, 7): 10, (2, 6, 7, 10): 10, (2, 6, 8): 4, (2, 4, 6, 8): 4, (2, 6, 9): 10, (2, 6, 9, 10): 10, (2, 6, 10): 8, (2, 6, 8, 10): 8, (2, 7, 8): 6, (2, 6, 7, 8): 6, (2, 7, 9): 6, (2, 6, 7, 9): 6, (2, 7, 10): 1, (1, 2, 7, 10): 1, (2, 8, 9): 6, (2, 6, 8, 9): 6, (2, 8, 10): 1, (1, 2, 8, 10): 1, (2, 9, 10): 3, (2, 3, 9, 10): 3, (3, 4, 5): 8, (3, 4, 5, 8): 8, (3, 4, 6): 7, (3, 4, 6, 7): 7, (3, 4, 7): 8, (3, 4, 7, 8): 8, (3, 4, 8): 2, (2, 3, 4, 8): 2, (3, 4, 9): 8, (3, 4, 8, 9): 8, (3, 4, 10): 8, (3, 4, 8, 10): 8, (3, 5, 6): 10, (3, 5, 6, 10): 10, (3, 5, 7): 2, (2, 3, 5, 7): 2, (3, 5, 8): 2, (2, 3, 5, 8): 2, (3, 5, 9): 2, (2, 3, 5, 9): 2, (3, 5, 10): 4, (3, 4, 5, 10): 4, (3, 6, 7): 10, (3, 6, 7, 10): 10, (3, 6, 8): 7, (3, 6, 7, 8): 7, (3, 6, 9): 10, (3, 6, 9, 10): 10, (3, 6, 10): 4, (3, 4, 6, 10): 4, (3, 7, 8): 2, (2, 3, 7, 8): 2, (3, 7, 9): 2, (2, 3, 7, 9): 2, (3, 7, 10): 4, (3, 4, 7, 10): 4, (3, 8, 9): 2, (2, 3, 8, 9): 2, (3, 8, 10): 7, (3, 7, 8, 10): 7, (3, 9, 10): 4, (3, 4, 9, 10): 4, (4, 5, 6): 9, (4, 5, 6, 9): 9, (4, 5, 7): 8, (4, 5, 7, 8): 8, (4, 5, 8): 2, (2, 4, 5, 8): 2, (4, 5, 9): 8, (4, 5, 8, 9): 8, (4, 5, 10): 8, (4, 5, 8, 10): 8, (4, 6, 7): 1, (1, 4, 6, 7): 1, (4, 6, 8): 1, (1, 4, 6, 8): 1, (4, 6, 9): 3, (3, 4, 6, 9): 3, (4, 6, 10): 2, (2, 4, 6, 10): 2, (4, 7, 8): 2, (2, 4, 7, 8): 2, (4, 7, 9): 8, (4, 7, 8, 9): 8, (4, 7, 10): 8, (4, 7, 8, 10): 8, (4, 8, 9): 2, (2, 4, 8, 9): 2, (4, 8, 10): 6, (4, 6, 8, 10): 6, (4, 9, 10): 8, (4, 8, 9, 10): 8, (5, 6, 7): 10, (5, 6, 7, 10): 10, (5, 6, 8): 9, (5, 6, 8, 9): 9, (5, 6, 9): 10, (5, 6, 9, 10): 10, (5, 6, 10): 4, (4, 5, 6, 10): 4, (5, 7, 8): 2, (2, 5, 7, 8): 2, (5, 7, 9): 2, (2, 5, 7, 9): 2, (5, 7, 10): 4, (4, 5, 7, 10): 4, (5, 8, 9): 2, (2, 5, 8, 9): 2, (5, 8, 10): 9, (5, 8, 9, 10): 9, (5, 9, 10): 4, (4, 5, 9, 10): 4, (6, 7, 8): 1, (1, 6, 7, 8): 1, (6, 7, 9): 10, (6, 7, 9, 10): 10, (6, 7, 10): 4, (4, 6, 7, 10): 4, (6, 8, 9): 3, (3, 6, 8, 9): 3, (6, 8, 10): 1, (1, 6, 8, 10): 1, (6, 9, 10): 4, (4, 6, 9, 10): 4, (7, 8, 9): 2, (2, 7, 8, 9): 2, (7, 8, 10): 1, (1, 7, 8, 10): 1, (7, 9, 10): 4, (4, 7, 9, 10): 4, (8, 9, 10): 3, (3, 8, 9, 10): 3}
```

```
s = input()
for _ in range(int(input())):
    n, k = map(int, input().split())
    nums = sorted(list(map(int, input().split())))
    if s == "add":
        if n == 1000000:
            print(228666)
        else:
            print(ansdict[tuple(nums)])
    else:
        if n == 1000000:
            for q in nums:
                if q != 228666:
                    print(q, end=" ")
            print()
        else:
            v = ansdict[tuple(nums)]
            for i in range(k + 1):
                if nums[i] != v:
                    print(nums[i], end=" ")
            print()
```

Task F ()

```
n = int(input())
print(4)
print(-3, -3)
print(-3, -1)
print(-1, -1)
print(-1, -3)
ans = [[-1, 2], [1, 2], [2, 0], [-2, 0], [-1, -2], [1, -2]]
for i in range(n):
    print(ans[i][0], ans[i][1])
```