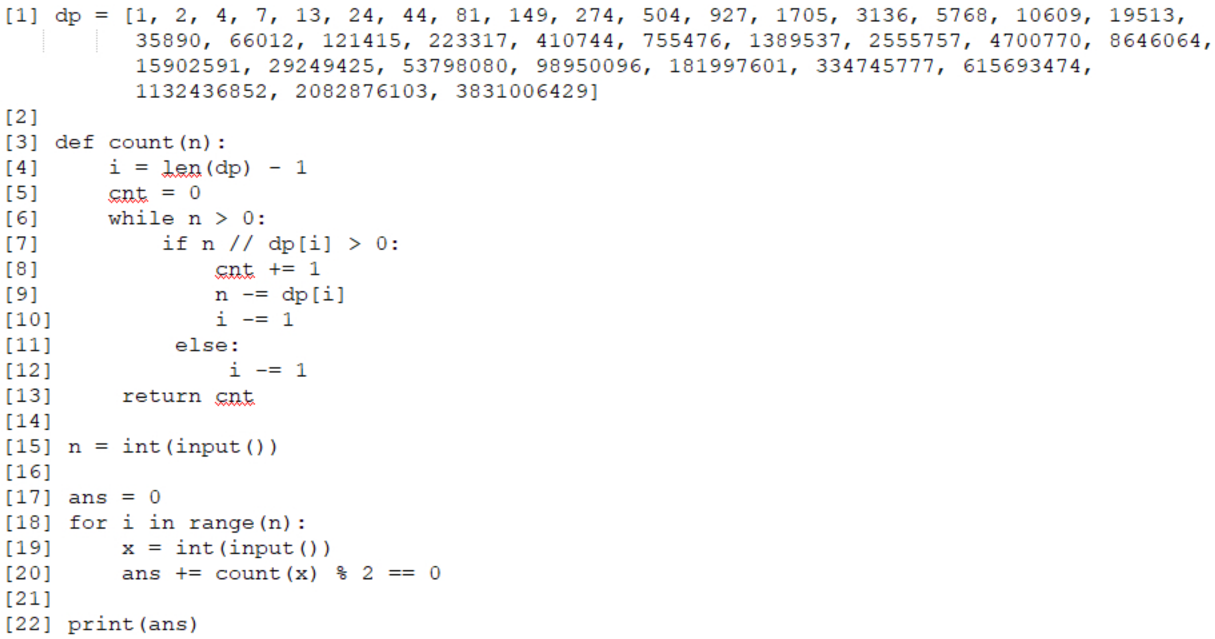
Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1206172, логином inf24f_229

Сводный итог по всем задачам в проверяющей системе

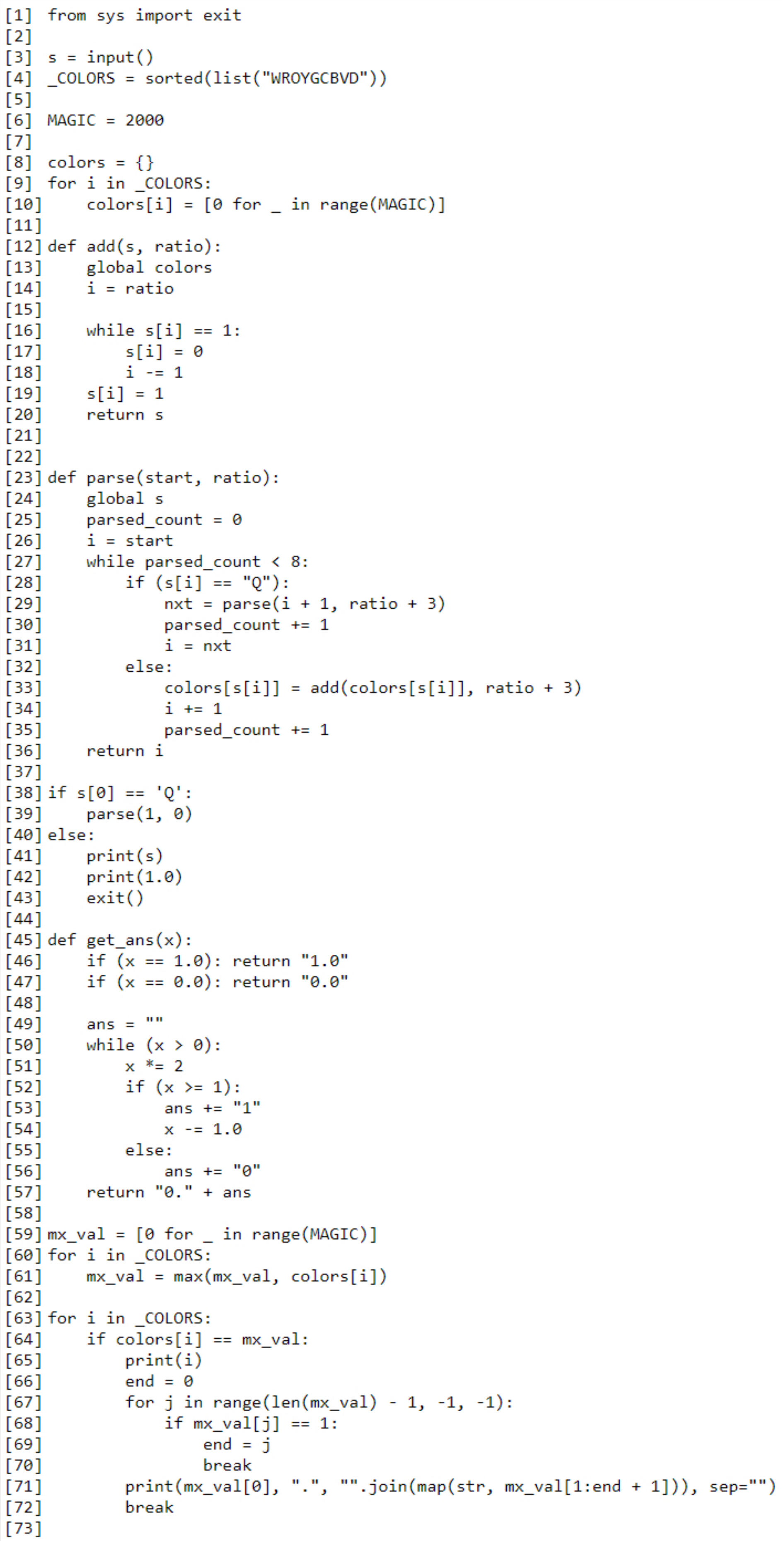
```
Run ID Time User name Problem Language Result Tests Score
494 3:17:08 inf24f_229 1 python3 Partial solution 7 16
479 3:12:58 inf24f_229 2 python3 OK 28 100
510 3:20:59 inf24f_229 3 python3 OK 28 100
621 3:50:06 inf24f_229 4 g++ Partial solution 3 10
270 1:49:58 inf24f_229 5 g++ OK 22 100
553 3:33:41 inf24f_229 6 python3 OK 11 100
426 технических баллов
71 итоговый балл
```

Посылка по задаче 1



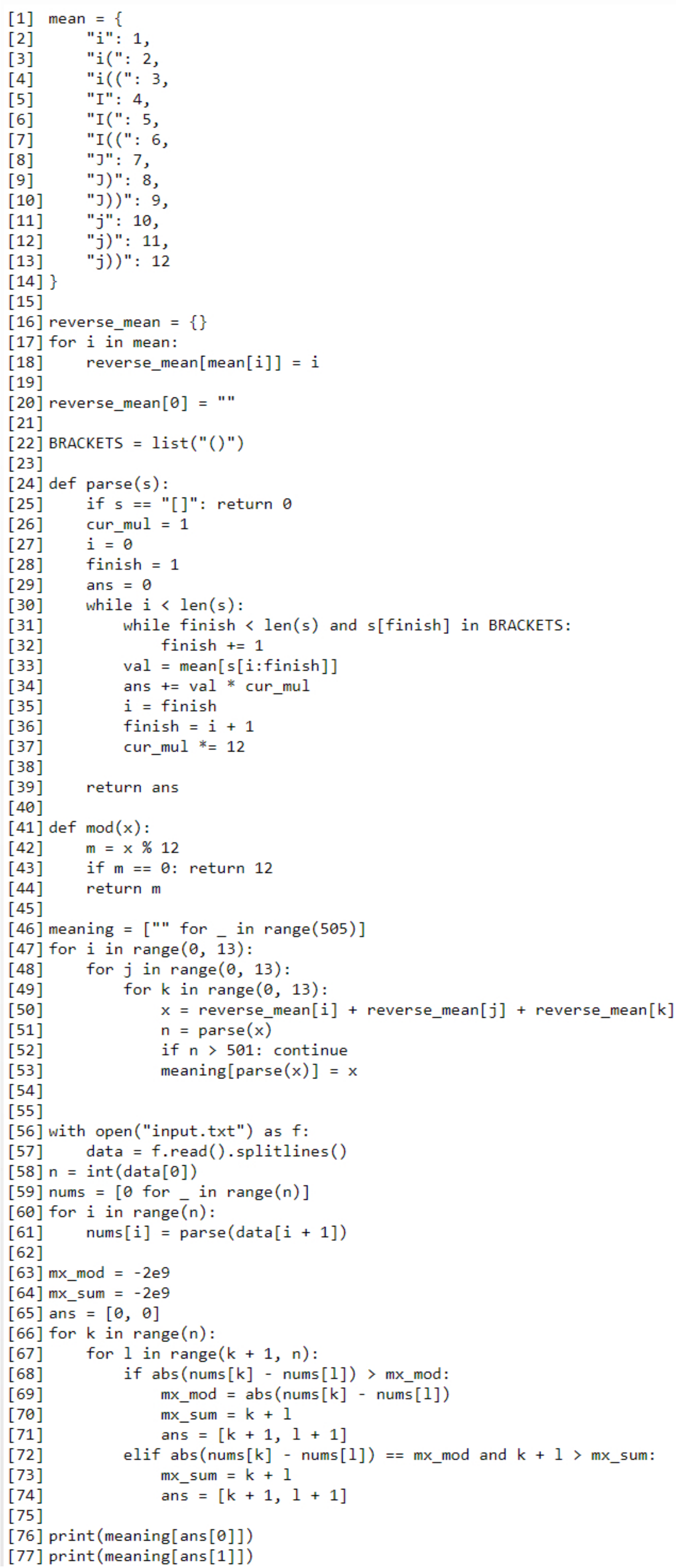
```
dp = [1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, 5768, 10609, 19513,
      35890, 66012, 121415, 223317, 410744, 755476, 1389537, 2555757, 4700770, 8646064,
      15902591, 29249425, 53798080, 98950096, 181997601, 334745777, 615693474,
      1132436852, 2082876103, 3831006429]

def count(n):
    i = len(dp) - 1
    cnt = 0
    while n > 0:
        if n // dp[i] > 0:
            cnt += 1
            n -= dp[i]
            i -= 1
        else:
            i -= 1
    return cnt

n = int(input())

ans = 0
for i in range(n):
    x = int(input())
    ans += count(x) % 2 == 0

print(ans)
```

Посылка по задаче 2



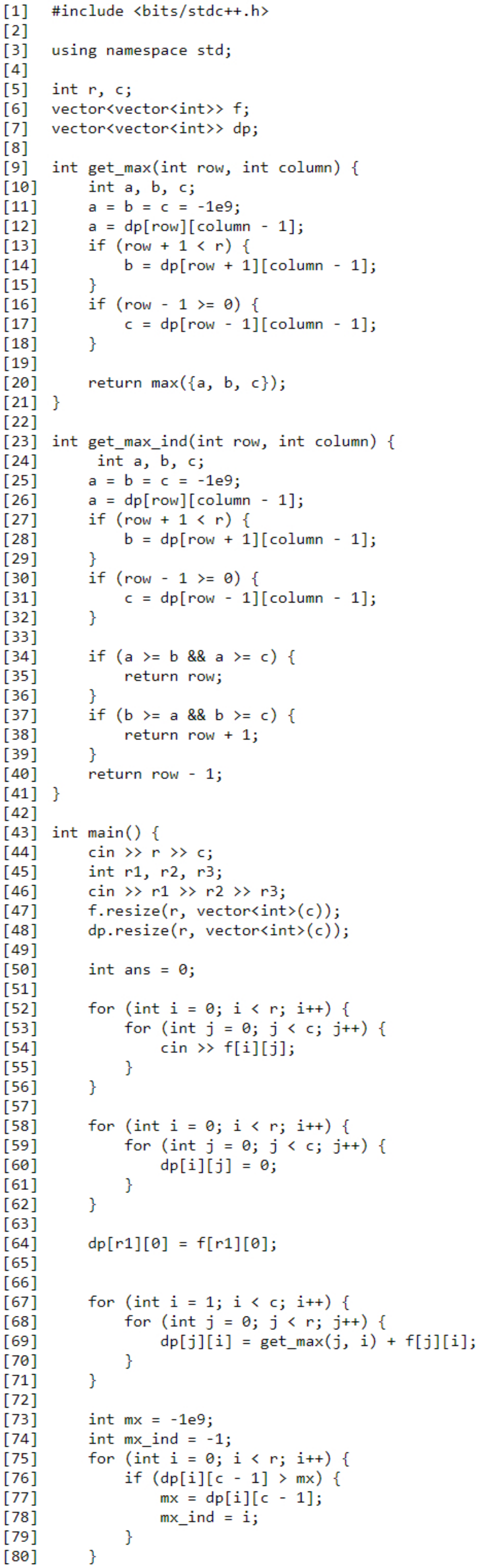
```python
from sys import exit

s = input()
_COLORS = sorted(list("WROYGCBVD"))

MAGIC = 2000

colors = {}
for i in _COLORS:
    colors[i] = [0 for _ in range(MAGIC)]

def add(s, ratio):
    global colors
    i = ratio

    while s[i] == 1:
        s[i] = 0
        i -= 1
    s[i] = 1
    return s


def parse(start, ratio):
    global s
    parsed_count = 0
    i = start
    while parsed_count < 8:
        if (s[i] == "Q"):
            nxt = parse(i + 1, ratio + 3)
            parsed_count += 1
            i = nxt
        else:
            colors[s[i]] = add(colors[s[i]], ratio + 3)
            i += 1
            parsed_count += 1
    return i

if s[0] == 'Q':
    parse(1, 0)
else:
    print(s)
    print(1.0)
    exit()

def get_ans(x):
    if (x == 1.0): return "1.0"
    if (x == 0.0): return "0.0"

    ans = ""
    while (x > 0):
        x *= 2
        if (x >= 1):
            ans += "1"
            x -= 1.0
        else:
            ans += "0"
    return "0." + ans

mx_val = [0 for _ in range(MAGIC)]
for i in _COLORS:
    mx_val = max(mx_val, colors[i])

for i in _COLORS:
    if colors[i] == mx_val:
        print(i)
        end = 0
        for j in range(len(mx_val) - 1, -1, -1):
            if mx_val[j] == 1:
                end = j
                break
        print(mx_val[0], ".", "".join(map(str, mx_val[1:end + 1])), sep="")
        break

```

Посылка по задаче 3



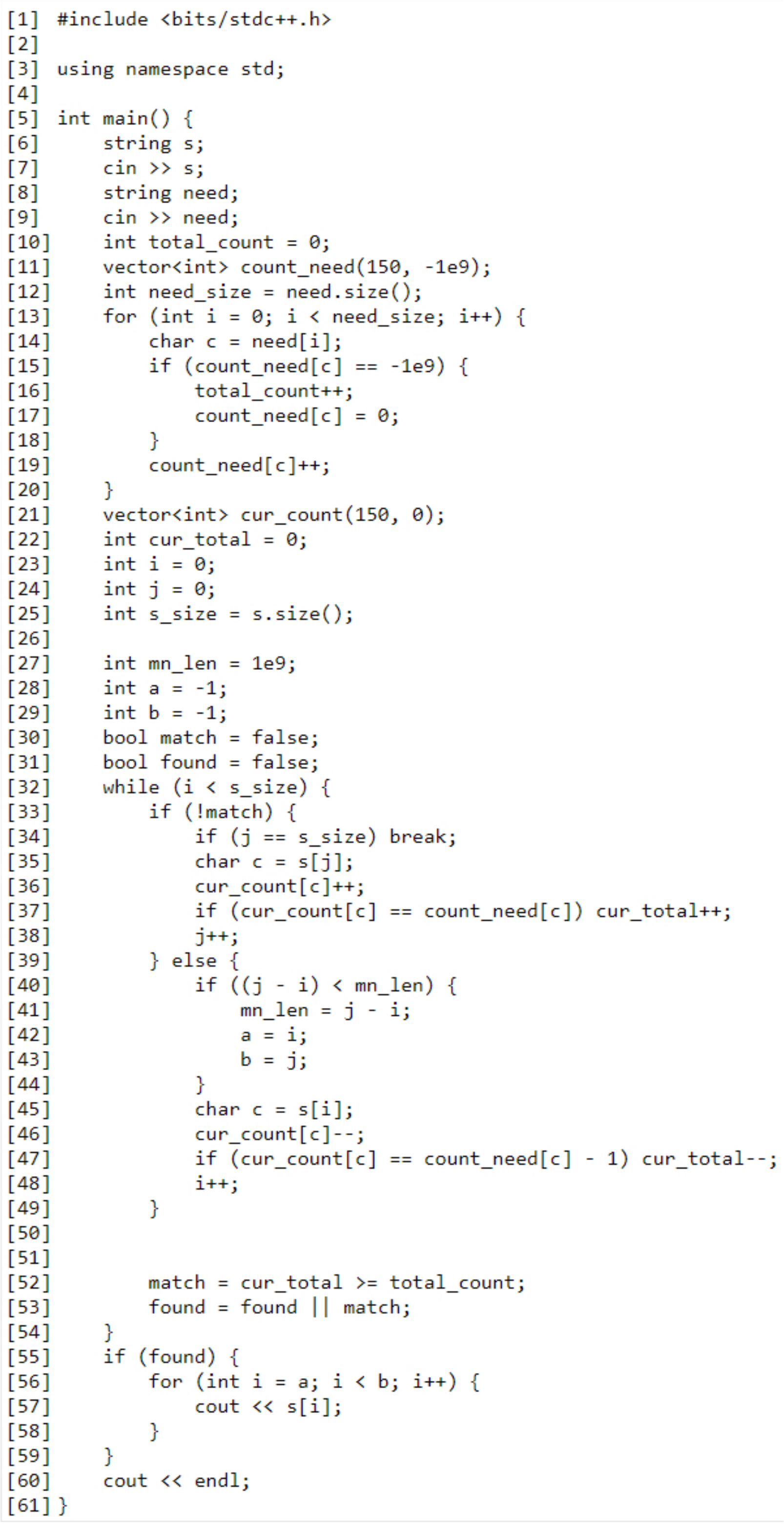
```
mean = {
    "i": 1,
    "i(": 2,
    "i((": 3,
    "I": 4,
    "I(": 5,
    "I((": 6,
    "J": 7,
    "J)": 8,
    "J))": 9,
    "j": 10,
    "j)": 11,
    "j))": 12
}

reverse_mean = {}
for i in mean:
    reverse_mean[mean[i]] = i

reverse_mean[0] = ""

BRACKETS = list("()")

def parse(s):
    if s == "[]": return 0
    cur_mul = 1
    i = 0
    finish = 1
    ans = 0
    while i < len(s):
        while finish < len(s) and s[finish] in BRACKETS:
            finish += 1
        val = mean[s[i:finish]]
        ans += val * cur_mul
        i = finish
        finish = i + 1
        cur_mul *= 12

    return ans

def mod(x):
    m = x % 12
    if m == 0: return 12
    return m

meaning = ["" for _ in range(505)]
for i in range(0, 13):
    for j in range(0, 13):
        for k in range(0, 13):
            x = reverse_mean[i] + reverse_mean[j] + reverse_mean[k]
            n = parse(x)
            if n > 501: continue
            meaning[parse(x)] = x


with open("input.txt") as f:
    data = f.read().splitlines()
n = int(data[0])
nums = [0 for _ in range(n)]
for i in range(n):
    nums[i] = parse(data[i + 1])

mx_mod = -2e9
mx_sum = -2e9
ans = [0, 0]
for k in range(n):
    for l in range(k + 1, n):
        if abs(nums[k] - nums[l]) > mx_mod:
            mx_mod = abs(nums[k] - nums[l])
            mx_sum = k + l
            ans = [k + 1, l + 1]
        elif abs(nums[k] - nums[l]) == mx_mod and k + l > mx_sum:
            mx_sum = k + l
            ans = [k + 1, l + 1]

print(meaning[ans[0]])
print(meaning[ans[1]])
```

Посылка по задаче 4



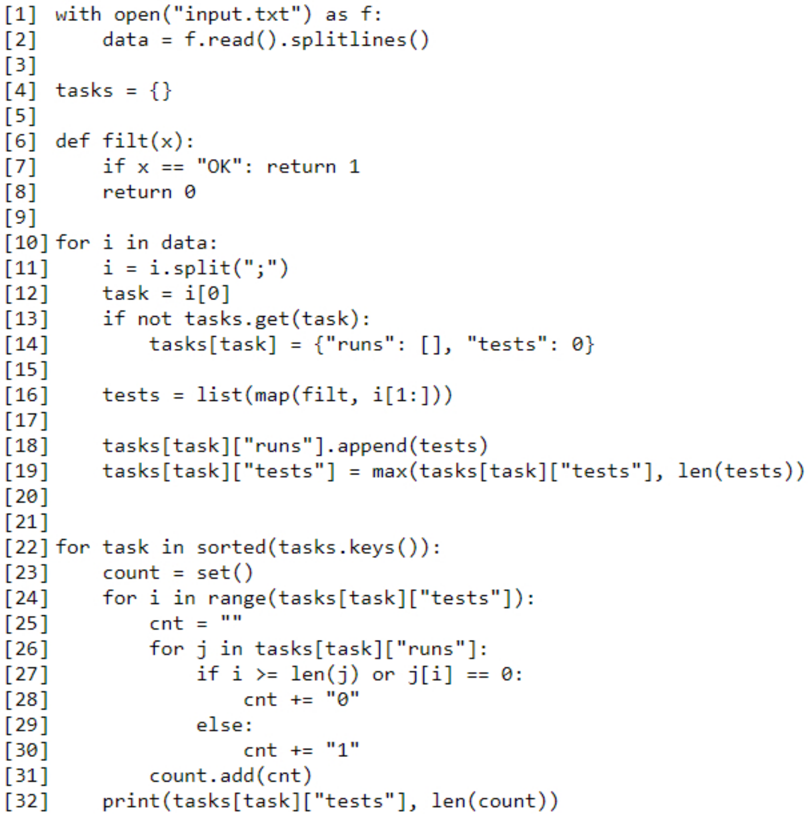
```cpp
#include <bits/stdc++.h>

using namespace std;

int r, c;
vector<vector<int>> f;
vector<vector<int>> dp;

int get_max(int row, int column) {
    int a, b, c;
    a = b = c = -1e9;
    a = dp[row][column - 1];
    if (row + 1 < r) {
        b = dp[row + 1][column - 1];
    }
    if (row - 1 >= 0) {
        c = dp[row - 1][column - 1];
    }

    return max({a, b, c});
}

int get_max_ind(int row, int column) {
     int a, b, c;
    a = b = c = -1e9;
    a = dp[row][column - 1];
    if (row + 1 < r) {
        b = dp[row + 1][column - 1];
    }
    if (row - 1 >= 0) {
        c = dp[row - 1][column - 1];
    }

    if (a >= b && a >= c) {
        return row;
    }
    if (b >= a && b >= c) {
        return row + 1;
    }
    return row - 1;
}

int main() {
    cin >> r >> c;
    int r1, r2, r3;
    cin >> r1 >> r2 >> r3;
    f.resize(r, vector<int>(c));
    dp.resize(r, vector<int>(c));

    int ans = 0;

    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            cin >> f[i][j];
        }
    }

    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            dp[i][j] = 0;
        }
    }

    dp[r1][0] = f[r1][0];


    for (int i = 1; i < c; i++) {
        for (int j = 0; j < r; j++) {
            dp[j][i] = get_max(j, i) + f[j][i];
        }
    }

    int mx = -1e9;
    int mx_ind = -1;
    for (int i = 0; i < r; i++) {
        if (dp[i][c - 1] > mx) {
            mx = dp[i][c - 1];
            mx_ind = i;
        }
    }
    ans += mx;
    int row = mx_ind;
    f[row][c - 1] = 0;
    for (int column = c - 1; column > 0; column--) {
        row = get_max_ind(row, column);
        f[row][column - 1] = 0;
    }


    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            dp[i][j] = 0;
        }
    }

    dp[r2][0] = f[r2][0];


    for (int i = 1; i < c; i++) {
        for (int j = 0; j < r; j++) {
            dp[j][i] = get_max(j, i) + f[j][i];
        }
    }

    mx = -1e9;
    mx_ind = -1;
    for (int i = 0; i < r; i++) {
        if (dp[i][c - 1] > mx) {
            mx = dp[i][c - 1];
            mx_ind = i;
        }
    }
    ans += mx;
    row = mx_ind;
    f[row][c - 1] = 0;
    for (int column = c - 1; column > 0; column--) {
        row = get_max_ind(row, column);
        f[row][column - 1] = 0;
    }



    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            dp[i][j] = 0;
        }
    }

    dp[r3][0] = f[r3][0];


    for (int i = 1; i < c; i++) {
        for (int j = 0; j < r; j++) {
            dp[j][i] = get_max(j, i) + f[j][i];
        }
    }

    mx = -1e9;
    mx_ind = -1;
    for (int i = 0; i < r; i++) {
        if (dp[i][c - 1] > mx) {
            mx = dp[i][c - 1];
            mx_ind = i;
        }
    }
    ans += mx;
    row = mx_ind;
    f[row][c - 1] = 0;
    for (int column = c - 1; column > 0; column--) {
        row = get_max_ind(row, column);
        f[row][column - 1] = 0;
    }

    cout << ans << endl;
}
```

Посылка по задаче 5



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

using namespace std;

int main() {
    string s;
    cin >> s;
    string need;
    cin >> need;
    int total_count = 0;
    vector<int> count_need(150, -1e9);
    int need_size = need.size();
    for (int i = 0; i < need_size; i++) {
        char c = need[i];
        if (count_need[c] == -1e9) {
            total_count++;
            count_need[c] = 0;
        }
        count_need[c]++;
    }
    vector<int> cur_count(150, 0);
    int cur_total = 0;
    int i = 0;
    int j = 0;
    int s_size = s.size();

    int mn_len = 1e9;
    int a = -1;
    int b = -1;
    bool match = false;
    bool found = false;
    while (i < s_size) {
        if (!match) {
            if (j == s_size) break;
            char c = s[j];
            cur_count[c]++;
            if (cur_count[c] == count_need[c]) cur_total++;
            j++;
        } else {
            if ((j - i) < mn_len) {
                mn_len = j - i;
                a = i;
                b = j;
            }
            char c = s[i];
            cur_count[c]--;
            if (cur_count[c] == count_need[c] - 1) cur_total--;
            i++;
        }


        match = cur_total >= total_count;
        found = found || match;
    }
    if (found) {
        for (int i = a; i < b; i++) {
            cout << s[i];
        }
    }
    cout << endl;
}
```

Посылка по задаче 6



```
with open("input.txt") as f:
    data = f.read().splitlines()

tasks = {}

def filt(x):
    if x == "OK": return 1
    return 0

for i in data:
    i = i.split(";")
    task = i[0]
    if not tasks.get(task):
        tasks[task] = {"runs": [], "tests": 0}

    tests = list(map(filt, i[1:]))

    tasks[task]["runs"].append(tests)
    tasks[task]["tests"] = max(tasks[task]["tests"], len(tests))


for task in sorted(tasks.keys()):
    count = set()
    for i in range(tasks[task]["tests"]):
        cnt = ""
        for j in tasks[task]["runs"]:
            if i >= len(j) or j[i] == 0:
                cnt += "0"
            else:
                cnt += "1"
        count.add(cnt)
    print(tasks[task]["tests"], len(count))
```