

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

```
[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]
[2]
[3] def count(n):
[4]     i = len(dp) - 1
[5]     cnt = 0
[6]     while n > 0:
[7]         if n // dp[i] > 0:
[8]             cnt += 1
[9]             n -= dp[i]
[10]            i -= 1
[11]        else:
[12]            i -= 1
[13]    return cnt
[14]
[15] n = int(input())
[16]
[17] ans = 0
[18] for i in range(n):
[19]     x = int(input())
[20]     ans += count(x) % 2 == 0
[21]
[22] print(ans)
```

Посылка по задаче 2

```
[1] from sys import exit
[2]
[3] s = input()
[4] _COLORS = sorted(list("WROYGCBVD"))
[5]
[6] MAGIC = 2000
[7]
[8] colors = {}
[9] for i in _COLORS:
[10]     colors[i] = [0 for _ in range(MAGIC)]
[11]
[12] def add(s, ratio):
[13]     global colors
[14]     i = ratio
[15]
[16]     while s[i] == 1:
[17]         s[i] = 0
[18]         i -= 1
[19]     s[i] = 1
[20]     return s
[21]
[22]
[23] def parse(start, ratio):
[24]     global s
[25]     parsed_count = 0
[26]     i = start
[27]     while parsed_count < 8:
[28]         if (s[i] == "Q"):
[29]             nxt = parse(i + 1, ratio + 3)
[30]             parsed_count += 1
[31]             i = nxt
[32]         else:
[33]             colors[s[i]] = add(colors[s[i]], ratio + 3)
[34]             i += 1
[35]             parsed_count += 1
[36]     return i
[37]
[38] if s[0] == 'Q':
[39]     parse(1, 0)
[40] else:
[41]     print(s)
[42]     print(1.0)
[43]     exit()
[44]
[45] def get_ans(x):
[46]     if (x == 1.0): return "1.0"
[47]     if (x == 0.0): return "0.0"
[48]
[49]     ans = ""
[50]     while (x > 0):
[51]         x *= 2
[52]         if (x >= 1):
[53]             ans += "1"
[54]             x -= 1.0
[55]         else:
[56]             ans += "0"
[57]     return "0." + ans
[58]
[59] mx_val = [0 for _ in range(MAGIC)]
[60] for i in _COLORS:
[61]     mx_val = max(mx_val, colors[i])
[62]
[63] for i in _COLORS:
[64]     if colors[i] == mx_val:
[65]         print(i)
[66]         end = 0
[67]         for j in range(len(mx_val) - 1, -1, -1):
[68]             if mx_val[j] == 1:
[69]                 end = j
[70]                 break
[71]         print(mx_val[0], ".", "".join(map(str, mx_val[1:end + 1])), sep="")
[72]         break
[73]
```

Посылка по задаче 3

```
[1] mean = {
[2]     "i": 1,
[3]     "i(": 2,
[4]     "i((": 3,
[5]     "I": 4,
[6]     "I(": 5,
[7]     "I((": 6,
[8]     "j": 7,
[9]     "j)": 8,
[10]    "j))": 9,
[11]    "j": 10,
[12]    "j)": 11,
[13]    "j))": 12
[14] }
[15]
[16] reverse_mean = {}
[17] for i in mean:
[18]     reverse_mean[mean[i]] = i
[19]
[20] reverse_mean[0] = ""
[21]
[22] BRACKETS = list("(")
[23]
[24] def parse(s):
[25]     if s == "[]": return 0
[26]     cur_mul = 1
[27]     i = 0
[28]     finish = 1
[29]     ans = 0
[30]     while i < len(s):
[31]         while finish < len(s) and s[finish] in BRACKETS:
[32]             finish += 1
[33]         val = mean[s[i:finish]]
[34]         ans += val * cur_mul
[35]         i = finish
[36]         finish = i + 1
[37]         cur_mul *= 12
[38]
[39]     return ans
[40]
[41] def mod(x):
[42]     m = x % 12
[43]     if m == 0: return 12
[44]     return m
[45]
[46] meaning = [""] for _ in range(505)]
[47] for i in range(0, 13):
[48]     for j in range(0, 13):
[49]         for k in range(0, 13):
[50]             x = reverse_mean[i] + reverse_mean[j] + reverse_mean[k]
[51]             n = parse(x)
[52]             if n > 501: continue
[53]             meaning[parse(x)] = x
[54]
[55]
[56] with open("input.txt") as f:
[57]     data = f.read().splitlines()
[58] n = int(data[0])
[59] nums = [0 for _ in range(n)]
[60] for i in range(n):
[61]     nums[i] = parse(data[i + 1])
[62]
[63] mx_mod = -2e9
[64] mx_sum = -2e9
[65] ans = [0, 0]
[66] for k in range(n):
[67]     for l in range(k + 1, n):
[68]         if abs(nums[k] - nums[l]) > mx_mod:
[69]             mx_mod = abs(nums[k] - nums[l])
[70]             mx_sum = k + 1
[71]             ans = [k + 1, l + 1]
[72]         elif abs(nums[k] - nums[l]) == mx_mod and k + 1 > mx_sum:
[73]             mx_sum = k + 1
[74]             ans = [k + 1, l + 1]
[75]
[76] print(meaning[ans[0]])
[77] print(meaning[ans[1]])
```

Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] int r, c;
[6] vector<vector<int>> f;
[7] vector<vector<int>> dp;
[8]
[9] int get_max(int row, int column) {
[10]     int a, b, c;
[11]     a = b = c = -1e9;
[12]     a = dp[row][column - 1];
[13]     if (row + 1 < r) {
[14]         b = dp[row + 1][column - 1];
[15]     }
[16]     if (row - 1 >= 0) {
[17]         c = dp[row - 1][column - 1];
[18]     }
[19]
[20]     return max({a, b, c});
[21] }
[22]
[23] int get_max_ind(int row, int column) {
[24]     int a, b, c;
[25]     a = b = c = -1e9;
[26]     a = dp[row][column - 1];
[27]     if (row + 1 < r) {
[28]         b = dp[row + 1][column - 1];
[29]     }
[30]     if (row - 1 >= 0) {
[31]         c = dp[row - 1][column - 1];
[32]     }
[33]
[34]     if (a >= b && a >= c) {
[35]         return row;
[36]     }
[37]     if (b >= a && b >= c) {
[38]         return row + 1;
[39]     }
[40]     return row - 1;
[41] }
[42]
[43] int main() {
[44]     cin >> r >> c;
[45]     int r1, r2, r3;
[46]     cin >> r1 >> r2 >> r3;
[47]     f.resize(r, vector<int>(c));
[48]     dp.resize(r, vector<int>(c));
[49]
[50]     int ans = 0;
[51]
[52]     for (int i = 0; i < r; i++) {
[53]         for (int j = 0; j < c; j++) {
[54]             cin >> f[i][j];
[55]         }
[56]     }
[57]
[58]     for (int i = 0; i < r; i++) {
[59]         for (int j = 0; j < c; j++) {
[60]             dp[i][j] = 0;
[61]         }
[62]     }
[63]
[64]     dp[r1][0] = f[r1][0];
[65]
[66]
[67]     for (int i = 1; i < c; i++) {
[68]         for (int j = 0; j < r; j++) {
[69]             dp[j][i] = get_max(j, i) + f[j][i];
[70]         }
[71]     }
[72]
[73]     int mx = -1e9;
[74]     int mx_ind = -1;
[75]     for (int i = 0; i < r; i++) {
[76]         if (dp[i][c - 1] > mx) {
[77]             mx = dp[i][c - 1];
[78]             mx_ind = i;
[79]         }
[80]     }
```

```

[81]     ans += mx;
[82]     int row = mx_ind;
[83]     f[row][c - 1] = 0;
[84]     for (int column = c - 1; column > 0; column--) {
[85]         row = get_max_ind(row, column);
[86]         f[row][column - 1] = 0;
[87]     }
[88]
[89]
[90]     for (int i = 0; i < r; i++) {
[91]         for (int j = 0; j < c; j++) {
[92]             dp[i][j] = 0;
[93]         }
[94]     }
[95]
[96]     dp[r2][0] = f[r2][0];
[97]
[98]
[99]     for (int i = 1; i < c; i++) {
[100]        for (int j = 0; j < r; j++) {
[101]            dp[j][i] = get_max(j, i) + f[j][i];
[102]        }
[103]    }
[104]
[105]    mx = -1e9;
[106]    mx_ind = -1;
[107]    for (int i = 0; i < r; i++) {
[108]        if (dp[i][c - 1] > mx) {
[109]            mx = dp[i][c - 1];
[110]            mx_ind = i;
[111]        }
[112]    }
[113]    ans += mx;
[114]    row = mx_ind;
[115]    f[row][c - 1] = 0;
[116]    for (int column = c - 1; column > 0; column--) {
[117]        row = get_max_ind(row, column);
[118]        f[row][column - 1] = 0;
[119]    }
[120]
[121]
[122]
[123]    for (int i = 0; i < r; i++) {
[124]        for (int j = 0; j < c; j++) {
[125]            dp[i][j] = 0;
[126]        }
[127]    }
[128]
[129]    dp[r3][0] = f[r3][0];
[130]
[131]
[132]    for (int i = 1; i < c; i++) {
[133]        for (int j = 0; j < r; j++) {
[134]            dp[j][i] = get_max(j, i) + f[j][i];
[135]        }
[136]    }
[137]
[138]    mx = -1e9;
[139]    mx_ind = -1;
[140]    for (int i = 0; i < r; i++) {
[141]        if (dp[i][c - 1] > mx) {
[142]            mx = dp[i][c - 1];
[143]            mx_ind = i;
[144]        }
[145]    }
[146]    ans += mx;
[147]    row = mx_ind;
[148]    f[row][c - 1] = 0;
[149]    for (int column = c - 1; column > 0; column--) {
[150]        row = get_max_ind(row, column);
[151]        f[row][column - 1] = 0;
[152]    }
[153]
[154]    cout << ans << endl;
[155] }

```

Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] int main() {
[6]     string s;
[7]     cin >> s;
[8]     string need;
[9]     cin >> need;
[10]    int total_count = 0;
[11]    vector<int> count_need(150, -1e9);
[12]    int need_size = need.size();
[13]    for (int i = 0; i < need_size; i++) {
[14]        char c = need[i];
[15]        if (count_need[c] == -1e9) {
[16]            total_count++;
[17]            count_need[c] = 0;
[18]        }
[19]        count_need[c]++;
[20]    }
[21]    vector<int> cur_count(150, 0);
[22]    int cur_total = 0;
[23]    int i = 0;
[24]    int j = 0;
[25]    int s_size = s.size();
[26]
[27]    int mn_len = 1e9;
[28]    int a = -1;
[29]    int b = -1;
[30]    bool match = false;
[31]    bool found = false;
[32]    while (i < s_size) {
[33]        if (!match) {
[34]            if (j == s_size) break;
[35]            char c = s[j];
[36]            cur_count[c]++;
[37]            if (cur_count[c] == count_need[c]) cur_total++;
[38]            j++;
[39]        } else {
[40]            if ((j - i) < mn_len) {
[41]                mn_len = j - i;
[42]                a = i;
[43]                b = j;
[44]            }
[45]            char c = s[i];
[46]            cur_count[c]--;
[47]            if (cur_count[c] == count_need[c] - 1) cur_total--;
[48]            i++;
[49]        }
[50]
[51]
[52]        match = cur_total >= total_count;
[53]        found = found || match;
[54]    }
[55]    if (found) {
[56]        for (int i = a; i < b; i++) {
[57]            cout << s[i];
[58]        }
[59]    }
[60]    cout << endl;
[61] }
```

Посылка по задаче 6

```
[1] with open("input.txt") as f:
[2]     data = f.read().splitlines()
[3]
[4] tasks = {}
[5]
[6] def filt(x):
[7]     if x == "OK": return 1
[8]     return 0
[9]
[10] for i in data:
[11]     i = i.split(";")
[12]     task = i[0]
[13]     if not tasks.get(task):
[14]         tasks[task] = {"runs": [], "tests": 0}
[15]
[16]     tests = list(map(filt, i[1:]))
[17]
[18]     tasks[task]["runs"].append(tests)
[19]     tasks[task]["tests"] = max(tasks[task]["tests"], len(tests))
[20]
[21]
[22] for task in sorted(tasks.keys()):
[23]     count = set()
[24]     for i in range(tasks[task]["tests"]):
[25]         cnt = ""
[26]         for j in tasks[task]["runs"]:
[27]             if i >= len(j) or j[i] == 0:
[28]                 cnt += "0"
[29]             else:
[30]                 cnt += "1"
[31]         count.add(cnt)
[32]     print(tasks[task]["tests"], len(count))
```