

Олимпиада «Ломоносов» по информатике  
2023-2024 учебный год. Заключительный тур  
Работа участника с id заявки 1240446, логином inf24f\_289

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
16	0:12:01	inf24f_289	1	g++	OK	28	100
122	0:54:34	inf24f_289	2	g++	OK	28	100
284	1:54:14	inf24f_289	3	python3	Partial solution	24	84
666	3:56:36	inf24f_289	4	g++	OK	21	100
593	3:44:30	inf24f_289	5	g++	OK	22	100
484 технических балла							
81 итоговый балл							

## Посылка по задаче 1

```
[1] #include <bits/stdc++.h>
[2]
[3] // #define int long long
[4] // #define x real()
[5] // #define y imag()
[6]
[7] using namespace std;
[8]
[9] //using point = complex<double>;
[10]
[11] int T[10000];
[12] void solve(){
[13]     T[0] = T[1] = T[2] = 0;
[14]     T[3] = 1;
[15]     const int mm = (1 << 24);
[16]     for(int i = 4; T[i - 1] < mm; ++i) {
[17]         T[i] = T[i - 1] + T[i - 2] + T[i - 3] + T[i - 4];
[18]     }
[19]     int n;
[20]     cin >> n;
[21]     int an = 0;
[22]     for(int i = 0; i < n; ++i){
[23]         int a;
[24]         cin >> a;
[25]         int cnt = 0;
[26]         int idx = 30;
[27]         while(a){
[28]             if(T[idx] <= a){
[29]                 a -= T[idx];
[30]                 ++cnt;
[31]             }
[32]             --idx;
[33]         }
[34]         if(cnt % 2 == 0){
[35]             ++an;
[36]         }
[37]     }
[38]     cout << an;
[39] }
[40]
[41]
[42] signed main() {
[43] //     ios_base::sync_with_stdio(false);
[44] //     cin.tie(nullptr);
[45]
[46]     int t;
[47] //     cin >> t;
[48]     t = 1;
[49]     for(int i = 0; i < t; ++i){
[50]         solve();
[51]     }
[52]
[53]     return 0;
[54] }
```

## Посылка по задаче 2

```
[1] #include <bits/stdc++.h>
[2]
[3] // #define int long long
[4]
[5] using namespace std;
[6]
[7] int mxdeep = 0;
[8] string s;
[9] int V[26][100000];
[10] int /*next idx*/ dfs(int idx, int deep){
[11]     idx += 1;
[12]     for(int i = 0; i < 8; ++i){
[13]         if(s[idx] == 'Q'){
[14]             idx = dfs(idx, deep + 1);
[15]         }else{
[16]             mxdeep = max(mxdeep, deep + 1);
[17]             ++idx;
[18]         }
[19]     }
[20]     return idx;
[21] }
[22] int dfs_c(int idx, int deep){
[23]     idx += 1;
[24]     for(int i = 0; i < 8; ++i){
[25]         if(s[idx] == 'Q'){
[26]             idx = dfs_c(idx, deep + 1);
[27]         }else{
[28]             V[s[idx] - 'A'][(deep + 1) * 3] += 1;
[29]             ++idx;
[30]         }
[31]     }
[32]     return idx;
[33] }
[34]
[35] void solve(){
[36]     cin >> s;
[37]     if(s.size() == 1){
[38]         cout << s << '\n';
[39]         cout << "1.0";
[40]         return;
[41]     }
[42]     dfs(0, 0);
[43]     dfs_c(0, 0);
[44]     int cntz0 = 0;
[45]     for(int i = 0; i < 26; ++i){
[46]         for(int j = 3 * (mxdeep + 1); j > 0; --j){
[47]             V[i][j - 1] += V[i][j] / 2;
[48]             V[i][j] %= 2;
[49]         }
[50]     }
[51]     int ans = 0;
[52]     for(int i = 1; i < 26; ++i){
[53]         for(int j = 0; j < 3 * (mxdeep + 1); ++j){
[54]             if(V[i][j] == V[ans][j]){
[55]                 continue;
[56]             }
[57]             if(V[i][j] > V[ans][j]){
[58]                 ans = i;
[59]             }
[60]             break;
[61]         }
[62]     }
[63]     for(int i = 0; i < 3 * (mxdeep + 1); ++i){
[64]         if(V[ans][i] > 0){
[65]             cntz0 ++;
[66]         }
[67]     }
[68]     cout << char(ans + 'A') << '\n';
[69]     cout << "0.";
[70]     for(int i = 1; cntz0; ++i){
[71]         if(V[ans][i] > 0){
[72]             cntz0 --;
[73]         }
[74]         cout << V[ans][i];
[75]     }
[76] }
[77]
[78] signed main() {
[79]     ios_base::sync_with_stdio(false);
[80]     cin.tie(nullptr);
[81]
[82]     int t;
[83] //     cin >> t;
[84]     t = 1;
[85]     for(int i = 0; i < t; ++i){
[86]         solve();
[87]     }
[88]
[89]     return 0;
[90] }
```

### Посылка по задаче 3

```
[1] n = int(input())
[2] arr = [0 for i in range(n)]
[3] st = [1 for _ in range(100)]
[4] for i in range(1, 100):
[5]     st[i] = 12 * st[i - 1]
[6]
[7]
[8] def elf_human(num: str):
[9]     lnt = len(num)
[10]    if num == '[]':
[11]        return 0
[12]    j = 0
[13]    an = 0
[14]    st = 1
[15]    ant = []
[16]    while j < lnt:
[17]        rer = True
[18]        if j + 2 < lnt:
[19]            if num[j + 1] == num[j + 2] == '(':
[20]                rer = False
[21]                if arr[j] == 'i':
[22]                    ant.append(3)
[23]            else:
[24]                ant.append(6)
[25]            j += 2
[26]        elif num[j + 1] == num[j + 2] == ')':
[27]            rer = False
[28]            if num[j] == 'j':
[29]                ant.append(12)
[30]            else:
[31]                ant.append(9)
[32]            j += 2
[33]        if rer and j + 1 < lnt:
[34]            if num[j + 1] == '(':
[35]                rer = False
[36]                if num[j] == 'i':
[37]                    ant.append(2)
[38]            else:
[39]                ant.append(5)
[40]            j += 1
[41]        elif num[j + 1] == ')':
[42]            rer = False
[43]            if num[j] == 'j':
[44]                ant.append(11)
[45]            else:
[46]                ant.append(8)
[47]            j += 1
[48]        if rer:
[49]            if num[j] == 'i':
[50]                ant.append(1)
[51]            elif num[j] == 'I':
[52]                ant.append(4)
[53]            elif num[j] == 'J':
[54]                ant.append(7)
[55]            elif num[j] == 'j':
[56]                ant.append(10)
[57]
[58]        j += 1
[59]    for i in ant:
[60]        an += i * st
[61]        st *= 12
[62]    return an
[63]
[64]
[65] def human_elf(num: int):
[66]     val = {
[67]         0: '[]',
[68]         1: 'i',
[69]         2: 'i(',
[70]         3: 'i(((',
[71]         4: 'I',
[72]         5: 'I(',
[73]         6: 'I(((',
[74]         7: 'J',
[75]         8: 'J)',
[76]         9: 'J))',
[77]         10: 'j',
[78]         11: 'j)',
[79]         12: 'j))',
[80]     }
```

```

[81]     if num == 0:
[82]         return ''
[83]     n12 = [0 for _ in range(100)]
[84]     last = -1
[85]     for i in range(99, -1, -1):
[86]         n12[i] = num // st[i]
[87]         if last == -1 and n12[i] != 0:
[88]             last = i
[89]             num %= st[i]
[90]     flag = False # 0 - null = 0, 1 - null > 0
[91]     an = []
[92]     for i in range(last + 1):
[93]         if n12[i] == 0:
[94]             if flag:
[95]                 an.append('j')
[96]             else:
[97]                 flag = True
[98]                 an.append('j'))'
[99]         else:
[100]            if flag:
[101]                if n12[i] == 1:
[102]                    if i == last:
[103]                        pass
[104]                    else:
[105]                        an.append('j'))'
[106]                else:
[107]                    flag = False
[108]                    an.append(val[n12[i] - 1])
[109]            else:
[110]                an.append(val[n12[i]])
[111]     return ''.join(an)
[112]
[113]
[114] for i in range(n):
[115]     t = input()
[116]     if t == '[':
[117]         arr[i] = 0
[118]     else:
[119]         arr[i] = elf_human(t)
[120]
[121] razni = -1
[122] suma = -1
[123] K = -1
[124] L = -1
[125] for i in range(n):
[126]     for j in range(i + 1, n):
[127]         stm = abs(arr[i] - arr[j])
[128]         if stm > razni:
[129]             sm = i + j + 2
[130]             K = i + 1
[131]             L = j + 1
[132]             suma = sm
[133]             razni = stm
[134]         elif stm == razni:
[135]             sm = i + j + 2
[136]             if sm > suma:
[137]                 K = i + 1
[138]                 L = j + 1
[139]                 suma = sm
[140]             razni = stm
[141] K, L = min(K, L), max(K, L)
[142] print(human_elf(K), human_elf(L), sep='\n')
[143]
[144]
[145]

```

## Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3] #define int long long
[4]
[5] using namespace std;
[6]
[7] const int K = 1550;
[8]
[9] map<int, int> mp;
[10] int dpsost[K];
[11] int dp[201][K];
[12] int arr[20][200];
[13]
[14] void solve() {
[15]     int r, C;
[16]     cin >> r >> C;
[17]     int r1, r2, r3;
[18]     cin >> r1 >> r2 >> r3;
[19]     for (int i = 0; i < r; ++i) {
[20]         for (int j = 0; j < C; ++j) {
[21]             cin >> arr[i][j];
[22]         }
[23]     }
[24]     for(int i = 0; i < C; ++ i){
[25]         for(int j = 0; j < K; ++j){
[26]             dp[i][j] = INT32_MIN;
[27]         }
[28]     }
[29]     int IDX = 0;
[30]     for (int i = 0; i < r; ++i) {
[31]         mp[(1 << i)] = IDX;
[32]         dpsost[IDX++] = (1 << i);
[33]     }
[34]     for (int i = 0; i < r; ++i) {
[35]         for (int j = i + 1; j < r; ++j) {
[36]             mp[(1 << i) | (1 << j)] = IDX;
[37]             dpsost[IDX++] = (1 << i) | (1 << j);
[38]             mp[(1 << i) | (1 << j) | (1 << 23) | (1 << 24)] = IDX;
[39]             dpsost[IDX++] = (1 << i) | (1 << j) | (1 << 23) | (1 << 24);
[40]         }
[41]     }
[42]     for (int i = 0; i < r; ++i) {
[43]         for (int j = i + 1; j < r; ++j) {
[44]             for (int k = j + 1; k < r; ++k) {
[45]                 mp[(1 << i) | (1 << j) | (1 << k)] = IDX;
[46]                 dpsost[IDX++] = (1 << i) | (1 << j) | (1 << k);
[47]             }
[48]         }
[49]     }
[50]     int MASK = (1 << r1) | (1 << r2) | (1 << r3);
[51]     if(__builtin_popcount(MASK) == 1){
[52]         dp[0][mp[MASK]] = arr[r1][0];
[53]     }else if(__builtin_popcount(MASK) == 2) {
[54]         int tt1 = r1 ^ r2 ^ r3; // x1
[55]         int tt2 = r1;
[56]         if(tt1 != r1){
[57]             tt2 = r1;
[58]         } else if(tt1 != r2){
[59]             tt2 = r2;
[60]         } else if(tt1 != r3){
[61]             tt2 = r3;
[62]         }
[63]         int RR = min({r2, r2, r3});
[64]         if(RR == tt2) {
[65]             dp[0][mp[MASK]] = arr[tt1][0] + arr[tt2][0];
[66]         }else{
[67]             dp[0][mp[MASK | (1 << 23) | (1 << 24)]] = arr[tt1][0] + arr[tt2][0];
[68]         }
[69]     }else{
[70]         dp[0][mp[MASK]] = arr[r1][0] + arr[r2][0]+ arr[r3][0];
[71]     }
[72]
[73]     for (int i = 1; i < C; ++i) {
[74]         for (int j = 0; j < IDX; ++j) {
[75]             int cnt = __builtin_popcount(dpsost[j]);
[76]             if (cnt == 1) {
[77]                 int pos = 0;
[78]                 for (int k = 0; k < 32; ++k) {
[79]                     if (dpsost[j] & (1 << k)) {
[80]                         pos = k;
[81]                         break;
[82]                     }
[83]                 }
[84]                 int posa, posb, posc;
[85]                 posa = posb = posc = pos;
[86]                 for (int a = -1; a <= 1; ++a) {
[87]                     for (int b = -1; b <= 1; ++b) {
[88]                         for (int c = -1; c <= 1; ++c) {
[89]                             int rpa = min(r - 1, max(0ll, posa + a));
[90]                             int rpb = min(r - 1, max(0ll, posb + b));
[91]                             int rpc = min(r - 1, max(0ll, posc + c));
[92]                             int mask = (1 << rpa) | (1 << rpb) | (1 << rpc);
[93]                             if (__builtin_popcount(mask) == 3) {
[94]                                 dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos][i]);

```

```

[95]         } else if (__builtin_popcount(mask) == 2) {
[96]             dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos][i]);
[97]             dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask | (1 << 23) |
[98] (1 << 24)]] + arr[pos][i]);
[99]         } else if (__builtin_popcount(mask) == 1) {
[100]             dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos][i]);
[101]         }
[102]     }
[103] }
[104] }
[105] } else if (cnt == 2) {
[106]     int pos1 = -1;
[107]     int pos2 = -1;
[108]     for (int k = 0; k < 32; ++k) {
[109]         if ((dpsost[j] & (1 << k)) && pos1 == -1) {
[110]             pos1 = k;
[111]         } else if (dpsost[j] & (1 << k)) {
[112]             pos2 = k;
[113]             break;
[114]         }
[115]     }
[116]     int posa, posb, posc;
[117]     posa = posb = pos1;
[118]     posc = pos2;
[119]     for (int a = -1; a <= 1; ++a) {
[120]         for (int b = -1; b <= 1; ++b) {
[121]             for (int c = -1; c <= 1; ++c) {
[122]                 int rpa = min(r - 1, max(0ll, posa + a));
[123]                 int rpb = min(r - 1, max(0ll, posb + b));
[124]                 int rpc = min(r - 1, max(0ll, posc + c));
[125]                 int mask = (1 << rpa) | (1 << rpb) | (1 << rpc);
[126]                 if (__builtin_popcount(mask) == 3) {
[127]                     dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1
[128] [i] + arr[pos2][i]);
[129]                 } else if (__builtin_popcount(mask) == 2) {
[130]                     int tt1 = rpa ^ rpb ^ rpc; // x1
[131]                     int tt2 = rpa;
[132]                     if (tt1 != rpa){
[133]                         tt2 = rpa;
[134]                     }
[135]                     if (tt1 != rpb){
[136]                         tt2 = rpb;
[137]                     }
[138]                     if (tt1 != rpc){
[139]                         tt2 = rpc;
[140]                     }
[141]                     if (tt1 > tt2) {
[142]                         dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1
[143] [i] + arr[pos2][i]);
[144]                     } else {
[145]                         dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask | (1 << 23) |
[146] (1 << 24)]] + arr[pos1][i] +
[147]                             arr[pos2][i]);
[148]                     }
[149]                 } else if (__builtin_popcount(mask) == 1) {
[150]                     dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1
[151] [i] + arr[pos2][i]);
[152]                 }
[153]             }
[154]         }
[155]     }
[156] } else if (cnt == 4) {
[157]     int pos1 = -1;
[158]     int pos2 = -1;
[159]     for (int k = 0; k < 32; ++k) {
[160]         if ((dpsost[j] & (1 << k)) && pos1 == -1) {
[161]             pos1 = k;
[162]         } else if (dpsost[j] & (1 << k)) {
[163]             pos2 = k;
[164]             break;
[165]         }
[166]     }
[167]     int posa, posb, posc;
[168]     posa = posb = pos2;
[169]     posc = pos1;
[170]     for (int a = -1; a <= 1; ++a) {
[171]         for (int b = -1; b <= 1; ++b) {
[172]             for (int c = -1; c <= 1; ++c) {
[173]                 int rpa = min(r - 1, max(0ll, posa + a));
[174]                 int rpb = min(r - 1, max(0ll, posb + b));
[175]                 int rpc = min(r - 1, max(0ll, posc + c));
[176]                 int mask = (1 << rpa) | (1 << rpb) | (1 << rpc);
[177]                 if (__builtin_popcount(mask) == 3) {
[178]                     dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1
[179] [i] + arr[pos2][i]);
[180]                 } else if (__builtin_popcount(mask) == 2) {
[181]                     int tt1 = rpa ^ rpb ^ rpc; // x1
[182]                     int tt2 = rpa;
[183]                     if (tt1 != rpa){
[184]                         tt2 = rpa;
[185]                     }
[186]                     if (tt1 != rpb){
[187]                         tt2 = rpb;
[188]                     }

```

```

[189]             if(tt1 != rpc){
[190]                 tt2 = rpc;
[191]             }
[192]             if (tt1 > tt2) {
[193]                 dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1]
[194] [i] + arr[pos2][i]);
[195]             } else {
[196]                 dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask | (1 << 23) |
[197] (1 << 24)]] + arr[pos1][i] +
[198]                             arr[pos2][i]);
[199]             }
[200]             } else if (__builtin_popcount(mask) == 1) {
[201]                 dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1]
[202] [i] + arr[pos2][i]);
[203]             }
[204]         }
[205]     }
[206] }
[207] } else if (cnt == 3) {
[208]     int pos1 = -1;
[209]     int pos2 = -1;
[210]     int pos3 = -1;
[211]     for (int k = 0; k < 32; ++k) {
[212]         if ((dpsost[j] & (1 << k)) && pos1 == -1) {
[213]             pos1 = k;
[214]         } else if ((dpsost[j] & (1 << k)) && pos2 == -1) {
[215]             pos2 = k;
[216]         } else if (dpsost[j] & (1 << k)) {
[217]             pos3 = k;
[218]             break;
[219]         }
[220]     }
[221]     int posa, posb, posc;
[222]     posa = pos2;
[223]     posc = pos1;
[224]     posb = pos3;
[225]     for (int a = -1; a <= 1; ++a) {
[226]         for (int b = -1; b <= 1; ++b) {
[227]             for (int c = -1; c <= 1; ++c) {
[228]                 int rpa = min(r - 1, max(0ll, posa + a));
[229]                 int rpb = min(r - 1, max(0ll, posb + b));
[230]                 int rpc = min(r - 1, max(0ll, posc + c));
[231]                 int mask = (1 << rpa) | (1 << rpb) | (1 << rpc);
[232]                 if (__builtin_popcount(mask) == 3) {
[233]                     dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1]
[234] [i] + arr[pos2][i] + arr[pos3][i]);
[235]                 } else if (__builtin_popcount(mask) == 2) {
[236]                     int tt1 = rpa ^ rpb ^ rpc; // x1
[237]                     int tt2 = rpa;
[238]                     if(tt1 != rpa){
[239]                         tt2 = rpa;
[240]                     }
[241]                     if(tt1 != rpb){
[242]                         tt2 = rpb;
[243]                     }
[244]                     if(tt1 != rpc){
[245]                         tt2 = rpc;
[246]                     }
[247]                     if (tt1 > tt2) {
[248]                         dp[i][j] = max(dp[i][j],
[249]                                     dp[i - 1][mp[mask]] + arr[pos1][i] + arr[pos2]
[250] [i] + arr[pos3][i]);
[251]                     } else {
[252]                         dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask | (1 << 23) |
[253] (1 << 24)]] + arr[pos1][i] +
[254]                                     arr[pos2][i] + arr[pos3][i]);
[255]                     }
[256]                 } else if (__builtin_popcount(mask) == 1) {
[257]                     dp[i][j] = max(dp[i][j], dp[i - 1][mp[mask]] + arr[pos1]
[258] [i] + arr[pos2][i] + arr[pos3][i]);
[259]                 }
[260]             }
[261]         }
[262]     }
[263] }
[264] }
[265] }
[266] int an = 0;
[267] for(int i = 0; i < K; ++i){
[268]     an = max(an, dp[C - 1][i]);
[269] }
[270] cout << an;
[271] }
[272] signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t;
    // cin >> t;
    t = 1;
    for(int i = 0; i < t; ++i){
        solve();
    }

    return 0;
}

```

## Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3] // #define int long long
[4]
[5] using namespace std;
[6] int bt[127];
[7] int at[127];
[8] int cntb = 0;
[9] void solve(){
[10]     string a, b;
[11]     cin >> a >> b;
[12]     for(auto& i : b){
[13]         bt[i] ++;
[14]         if(bt[i] == 1){
[15]             ++cntb;
[16]         }
[17]     }
[18]     int i = 0, j = 0;
[19]     int cnt = 0;
[20]     int ib = -1, jb = (int)a.size()+178;
[21]     for(; i < a.size(); ++i){
[22]         while(j < a.size() && cnt < cntb){
[23]             ++at[a[j]];
[24]             if(at[a[j]] == bt[a[j]]){
[25]                 ++cnt;
[26]             }
[27]             ++j;
[28]         }
[29]         if(ib == -1 && cnt != cntb){
[30]             cout << " ";
[31]             return;
[32]         }
[33]         else if(cnt != cntb){
[34]             break;
[35]         }
[36]         if(jb - ib > j - i){
[37]             ib = i;
[38]             jb = j;
[39]         }
[40]         --at[a[i]];
[41]         if(at[a[i]] < bt[a[i]]){
[42]             --cnt;
[43]         }
[44]     }
[45]     for(int i_ = ib; i_ < jb; ++i_){
[46]         cout << a[i_];
[47]     }
[48] }
[49]
[50] signed main() {
[51]     ios_base::sync_with_stdio(false);
[52]     cin.tie(nullptr);
[53]
[54]     int t;
[55] //     cin >> t;
[56]     t = 1;
[57]     for(int i = 0; i < t; ++i){
[58]         solve();
[59]     }
[60]
[61]     return 0;
[62] }
```

Посылоч по задаче 6 не было