

Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1223989, логином inf24f_296

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
20	0:12:54	inf24f_296	1	g++	OK	28	100
83	0:36:30	inf24f_296	2	g++	Partial solution	27	96
552	3:33:14	inf24f_296	3	python3	OK	28	100
552	3:33:14	inf24f_296	3	python3	OK	28	100
653	3:54:51	inf24f_296	5	g++	Partial solution	3	5
505	3:19:40	inf24f_296	6	python3	Partial solution	10	90
491	технический балл						
81	итоговый балл						

Посылка по задаче 1

```
[1] #include <iostream>
[2] #include <vector>
[3]
[4] using namespace std;
[5]
[6] vector<int> t;
[7]
[8] int tetracount(int x)
[9] {
[10]     int ans =0;
[11]     for(int i=t.size()-1;i>-1;i--)
[12]     {
[13]         if(x==0)
[14]             return ans;
[15]         if(t[i] <= x)
[16]         {
[17]             //cout<<t[i]<<endl;
[18]             ans ++;
[19]             x-= t[i];
[20]         }
[21]     }
[22]     return ans;
[23] }
[24]
[25] int main()
[26] {
[27]     const int max_num = 33554432;
[28]     t.resize(4);
[29]     t[0] = 0;
[30]     t[1] = 0;
[31]     t[2] = 0;
[32]     t[3] = 1;
[33]
[34]     int x = 0;
[35]     while(x < max_num)
[36]     {
[37]         int n = t.size();
[38]         x = t[n-1] + t[n-2] + t[n-3] + t[n-4];
[39]         t.push_back(x);
[40]     }
[41]
[42]     //for(int i=0;i<t.size();i++)
[43]     //    cout<<t[i]<<' ';
[44]     //cout<<endl;
[45]
[46]     int n;
[47]     cin>>n;
[48]     int ans =0;
[49]     for(int i=0;i<n;i++)
[50]     {
[51]         int a;
[52]         cin>>a;
[53]         if(tetracount(a) % 2 ==0)
[54]             ans ++;
[55]     }
[56]     cout<<ans;
[57]
[58]     return 0;
[59] }
```

Посылка по задаче 2

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <string>
[4]
[5]
[6] using namespace std;
[7]
[8] struct bin_add
[9] {
[10]     vector<bool> x;
[11]     double value=0;
[12]
[13]     void add(int pos,double val)
[14]     {
[15]         //cout<<pos<<' '<<val<<endl;
[16]         value += val;
[17]         if(pos+1 > x.size())
[18]         {
[19]             int oldsize = x.size();
[20]             x.resize(pos+1);
[21]             for(int i=oldsize;i<x.size();i++)
[22]                 x[i] = false;
[23]         }
[24]         while(x[pos])
[25]         {
[26]
[27]             x[pos] = false;
[28]             pos--;
[29]         }
[30]         x[pos] = true;
[31]     }
[32]
[33]     string print()
[34]     {
[35]         if(x[0])
[36]             return "1.0";
[37]         string ans = "0.";
[38]
[39]         int border = x.size();
[40]         for(int i=x.size()-1;i>-1;i--)
[41]             if(x[i])
[42]             {
[43]                 border = i+1;
[44]                 break;
[45]             }
[46]         for(int i=1;i<border;i++)
[47]         {
[48]             if(x[i])
[49]                 ans+="1";
[50]             else
[51]                 ans+="0";
[52]         }
[53]         return ans;
[54]     }
[55] };
```

```

[56]
[57] vector<bin_add> k;
[58]
[59] string conv = "BCDGORVWY";
[60]
[61] int read(string& input,int x,int depth,double val)
[62] {
[63]     if(input[x] != 'Q')
[64]     {
[65]         for(int i=0;i<conv.size();i++)
[66]             if(conv[i] == input[x])
[67]                 k[i].add(depth,val);
[68]         return x+1;
[69]     }
[70]
[71]     x++;
[72]     for(int i=0;i<8;i++)
[73]         x = read(input,x,depth+3,val/8);
[74]     return x;
[75] }
[76]
[77]
[78] int main()
[79] {
[80]     k.resize(9);
[81]
[82]     string x;
[83]     cin>>x;
[84]     read(x,0,0,1);
[85]
[86]     int max_pos = 0;
[87]     for(int i=0;i<k.size();i++)
[88]     {
[89]         if(k[i].value > k[max_pos].value)
[90]             max_pos = i;
[91]     }
[92]
[93]     cout<<conv[max_pos]<<'\n';
[94]     cout<<k[max_pos].print();
[95]     return 0;
[96] }

```

Посылка по задаче 3

```
[1] from random import randint
[2] def parse_elf(x):
[3]     # print("PARSING",x)
[4]     if x=="[]":
[5]         return 0
[6]
[7]     n = len(x)
[8]     ans = 0
[9]     mul = 1
[10]    pos = 0
[11]
[12]    while pos < n:
[13]        digit = 0
[14]        if x[pos] == 'i':
[15]            digit = 1
[16]        if x[pos] == 'I':
[17]            digit = 4
[18]        if x[pos] == 'J':
[19]            digit = 7
[20]        if x[pos] == 'j':
[21]            digit = 10
[22]
[23]
[24]        stv = '('
[25]        if x[pos] == 'j' or x[pos] == 'J':
[26]            stv = ')'
[27]
[28]
[29]        if pos+1 < n:
[30]            if x[pos+1] == stv:
[31]                digit += 1
[32]                pos += 1
[33]        if pos+1 < n:
[34]            if x[pos+1] == stv:
[35]                digit += 1
[36]                pos += 1
[37]
[38]        pos += 1
[39]        # print("DIGIT: ",digit,ans,mul)
[40]        ans += digit * mul
[41]        mul = mul * 12
[42]    # print(x,ans)
[43]    return ans
[44]
[45]
[46] digits = ['0','i','i','i','i(','I','I(','I(','J','J'),'J'),'j','j'),'j)']
[47]
[48] def to_elf(a):
[49]     ans = ''
[50]     mul = 1
[51]     while a > 0:
[52]         digit = a % 12
[53]         if digit == 0:
[54]             digit = 12
[55]         a -= 12
[56]         ans = ans + digits[digit]
[57]         a = a // 12
[58]         # mul = mul * 12
[59]     return ans
[60]
[61] n = 0
[62] n = int(input())
[63]
[64] values = [0] * n
[65] sl = [0] * n
[66]
[67] for i in range(n):
[68]     sl[i] = input()
[69]     values[i] = parse_elf(sl[i])
[70]
[71] max_pos = 0
[72] min_pos = 1
[73]
[74] for i in range(n):
[75]     if(values[max_pos] <= values[i]):
[76]         max_pos = i
[77] for i in range(n):
[78]     if(values[min_pos] >= values[i] and i!=max_pos):
[79]         min_pos = i
[80]
[81] if max_pos < min_pos:
[82]     max_pos,min_pos = min_pos,max_pos
[83]
[84]
[85]
[86] print(to_elf(min_pos+1))
[87] print(to_elf(max_pos+1))
[88]
```

Посылка по задаче 4

```
[1] #include <iostream>
[2] #include <vector>
[3]
[4] using namespace std;
[5]
[6] const int UND = -5;
[7]
[8] pair<int,int> determine(pair<int,int> current,pair<int,int> new_p,int pos,int add)
[9] {
[10]     if(new_p.first == UND)
[11]         return current;
[12]     if(current.first == UND)
[13]         return {new_p.first+add,pos};
[14]     if(current.first < new_p.first+add)
[15]         return {new_p.first+add,pos};
[16]     return current;
[17] }
[18]
[19] vector<vector<int> > field;
[20] int r,c;
[21]
[22] int sum(int h,int a,int b,int c)
[23] {
[24]     int ans = field[a][h];
[25]     if(b!=a)
[26]         ans += field[b][h];
[27]     if(c!=a && c!=b)
[28]         ans += field[c][h];
[29]     return ans;
[30] }
[31]
[32] bool valid(int a,int b,int c)
[33] {
[34]     return (a > -1) && (b > -1) && (c > -1) && (a < r) && (b < r) && (c < r);
[35] }
[36]
[37] int main()
[38] {
[39]
[40]     cin>>r>>c;
[41]     int p1,p2,p3;
[42]     cin>>p1>>p2>>p3;
[43]
[44]     vector<int> starts = {p1,p2,p3};
[45]
[46]
[47]     field.resize(r);
[48]     for(int i=0;i<r;i++)
[49]         field[i].resize(c);
[50]
[51]     for(int i =0;i<r;i++)
[52]         for(int j =0;j<c;j++)
[53]             cin>>field[i][j];
[54]
[55]
[56]
```

```

[51]     for(int i =0;i<r;i++)
[52]         for(int j =0;j<c;j++)
[53]             cin>>field[i][j];
[54]
[55]
[56]
[57]     vector<vector<vector<vector<int> > > > dyn;
[58]     dyn.resize(c);
[59]     for(int i=0;i<c;i++)
[60]         dyn[i].resize(r);
[61]     for(int i=0;i<c;i++)
[62]         for(int j=0;j<r;j++)
[63]             dyn[i][j].resize(r);
[64]     for(int i=0;i<c;i++)
[65]         for(int j=0;j<r;j++)
[66]             for(int k=0;k<r;k++)
[67]                 dyn[i][j][k].resize(r);
[68]     for(int i=0;i<c;i++)
[69]         for(int j=0;j<r;j++)
[70]             for(int k=0;k<r;k++)
[71]                 for(int l=0;l<r;l++)
[72]                     dyn[i][j][k][l] = UND;
[73]
[74]
[75]     dyn[0][p1][p2][p3] = sum(0,p1,p2,p3);
[76]
[77]     for(int i =0;i<c-1;i++)
[78]     {
[79]         for(int j=0;j<r;j++)
[80]             for(int k=0;k<r;k++)
[81]                 for(int l=0;l<r;l++)
[82]                     if(dyn[i][j][k][l] != UND)
[83]                         for(int add1=-1;add1<2;add1++)
[84]                             for(int add2=-1;add2<2;add2++)
[85]                                 for(int add3=-1;add3<2;add3++)
[86]                                     if(valid(j+add1,k+add2,l+add3))
[87]                                         dyn[i+1][j+add1][k+add2][l+add3] = max(dyn[i+1][j+add1][k+add2][l+add3],dyn[i][j][k][l] + sum(i+1,j+add1,k+add2,l+add3));
[88]     }
[89]
[90]
[91]     //for(int i=0;i<c;i++)
[92]     int summ = 0;
[93]     for(int j=0;j<r;j++)
[94]         for(int k=0;k<r;k++)
[95]             for(int l=0;l<r;l++)
[96]                 summ = max(summ,dyn[c-1][j][k][l]);
[97]         //cout<<i<<','<<j<<','<<k<<','<<l<<": "<<dyn[i][j][k][l]<<'\n';
[98]
[99]
[100]     cout<<summ<<'\n';
[101]
[102]     return 0;
[103] }

```

Посылка по задаче 5

```
[1] #include <iostream>
[2] #include <set>
[3] #include <vector>
[4] #include <map>
[5]
[6] using namespace std;
[7]
[8]
[9]
[10] int main()
[11] {
[12]     string have;
[13]     string required;
[14]     cin>>have;
[15]     cin>>required;
[16]
[17]     map<int,int> task;
[18]     //cout<<(task.find(0) == task.end());
[19]
[20]     for(int i=0;i<required.size();i++)
[21]         task[int(required[i])]++;
[22]
[23]
[24]
[25]     vector<int> dyn;
[26]     dyn.resize(have.size());
[27]
[28]     int reqs = required.size();
[29]     for(int i =0;i<have.size();i++)
[30]     {
[31]         if(task.find(int(have[i])) != task.end())
[32]         {
[33]             task[int(have[i])] --;
[34]             if(task[int(have[i])] == 0)
[35]                 reqs --;
[36]         }
[37]         if(reqs == 0)
[38]         {
[39]             dyn[0] = i+1;
[40]             break;
[41]         }
[42]     }
[43]
[44]     if(reqs != 0)
[45]     {
[46]         cout<<" ";
[47]         return 0;
[48]     }
[49]
```



```

[50]     for(int i=1;i<have.size();i++)
[51]         dyn[i] = -1;
[52]
[53]     for(int i = 0;i<have.size()-1;i++)
[54]     {
[55]         //cout<<"POSITION:"<<i<<endl;
[56]         if(task.find(int(have[i])) != task.end())
[57]         {
[58]             //cout<<"NEEDED:"<<have[i]<<' '<<task[int(have[i])]<<endl;
[59]             task[int(have[i])] ++;
[60]             if(task[int(have[i])] > 0)
[61]             {
[62]                 //cout<<"OUT OF:"<<have[i]<<endl;
[63]                 dyn[i+1] = -1;
[64]                 for(int j =i+dyn[i-1] - 1;j<have.size();j++)
[65]                 {
[66]                     if(task.find(int(have[j])) != task.end())
[67]                         task[int(have[j])] --;
[68]                     if(int(have[j]) ==int(have[i]))
[69]                     {
[70]                         //cout<<"FOUND:"<<j<<endl;
[71]                         dyn[i+1] = j - i;
[72]                         break;
[73]                     }
[74]                 }
[75]                 if(dyn[i+1] ==-1)
[76]                 {
[77]                     dyn[i+1] = -1;
[78]                     break;
[79]                 }
[80]
[81]             }else
[82]                 dyn[i+1] = dyn[i] - 1;
[83]         }else
[84]             dyn[i+1] = dyn[i] - 1;
[85]     }
[86]
[87]
[88]     int min_len = 0;
[89]     for(int i=0;i<have.size();i++)
[90]         if(dyn[i] < dyn[min_len] && dyn[i] != -1)
[91]             min_len = i;
[92]
[93]     for(int i=min_len;i<min_len+dyn[min_len];i++)
[94]         cout<<have[i];
[95]
[96]
[97]
[98]     return 0;
[99] }

```

Посылка по задаче 6

```
[1] f = open("input.txt", 'r')
[2]
[3] tests = f.read().split('\n')
[4]
[5] f.close()
[6]
[7] for i in range(len(tests)):
[8]     tests[i] = tests[i].split(';')
[9]
[10] for i in range(len(tests)):
[11]     for j in range(1, len(tests[i])):
[12]         tests[i][j] = (tests[i][j] == 'OK')
[13]
[14] tasks = dict()
[15]
[16] for i in range(len(tests)):
[17]     if tests[i][0] in tasks:
[18]         tasks[tests[i][0]].append(tests[i][1:])
[19]     else:
[20]         tasks[tests[i][0]] = [(tests[i][1:])]
[21]
[22] for k in tasks.keys():
[23]     max_len = 0
[24]     for i in range(len(tasks[k])):
[25]         max_len = max(max_len, len(tasks[k][i]))
[26]     for i in range(len(tasks[k])):
[27]         while(len(tasks[k][i]) != max_len):
[28]             tasks[k][i].append(False)
[29]
[30] # print(tasks)
[31]
[32] for k in tasks.keys():
[33]     gh = set()
[34]     for t in range(len(tasks[k][0])):
[35]         test_id = 0
[36]         for i in range(len(tasks[k])):
[37]             if tasks[k][i][t]:
[38]                 test_id = test_id * 2 + 1
[39]             else:
[40]                 test_id = test_id * 2
[41]         # print(test_id)
[42]         gh.add(test_id)
[43]     # ans_k[k] = len(gh)
[44]     if(k != ''):
[45]         print(len(tasks[k][0]), len(gh))
```