

Гальченко Дмитрий Геннадьевич

Total = 210

Compiler:

Compiler - GNU C++17 7.2.0 (msys2 mingw-w64-x86_64)

Задача А. Эпическая потасовка

Ответ участника:

```
1 //Problem A (100)
2 #include <bits/stdc++.h>
3
4 using namespace std;
5
6 typedef long long ll;
7
8 signed main()
9 {
10  ll h1, a1, h2, a2;
11  cin >> h1 >> a1 >> h2 >> a2;
12  ll cnt1 = (h2 + a1 - 1) / a1;
13  ll cnt2 = (h1 + a2 - 1) / a2;
14  /* cout << cnt1 << " " << cnt2 << "\n";*/
15  if (cnt1 <= cnt2)
16  {
17    cout << "Bandergolf Cyberswitch\n";
18  }
19  else
20  {
21    cout << "Benadryl Cabbagepatch\n";
22  }
23  return 0;
24 }
```

Задача В. Весы и монеты

Ответ участника:

```
1 //Problem B (100)
2 #include <bits/stdc++.h>
3
4 using namespace std;
5
6 typedef unsigned long long ll;
7
8 ll n, k;
9
10 signed main()
11 {
12     cin >> k >> n;
13     ll res = 0;
14     for (ll q = 1; q < n; q *= (k + 1)) {
15         res++;
16     }
17     cout << res << "\n";
18     return 0;
19 }
```

Задача С. Индийская клавиатура

Ответ участника :

Задача D. Киви

Ответ участника:

```
1 //Problem D (10)
2 #include <bits/stdc++.h>
3
4 using namespace std;
5
6 const int N = 1e5;
7
8 int n;
9 vector<int> g[N];
10 bool is_root[N], is_leaf[N];
11
12 int get_root() {
13     for (int i = 0; i < n; i++) {
14         if (is_root[i]) {
15             return i;
16         }
17     }
18 }
19
20 void solve4(int r, bool ok) {
21     if (g[r].size() == 3) {
22         cout << 2 + ok << " 2 2 1";
23     } else if (g[r].size() == 2) {
24         cout << 3 + ok << " 2 1 1";
25     } else if (g[g[r][0]].size() == 2) {
26         cout << 3 + ok << " 2 1 1";
27     } else {
28         cout << 4 + ok << " 1 1 1";
29     }
30 }
31
32 signed main() {
33     cin >> n;
34     fill(is_root, is_root + n, true);
35     fill(is_leaf, is_leaf + n, true);
36     int ct = 0;
37     for (int i = 0; i < n - 1; i++) {
38         int v, u;
39         cin >> v >> u;
40         v--;
41         u--;
42         ct += v + 1 == u;
43         g[v].emplace_back(u);
44         is_leaf[v] = false;
45         is_root[u] = false;
46     }
47     if (ct == n - 1) {
48         cout << n;
49         for (int i = 0; i < n - 1; i++) {
50             cout << " 1";
51         }
52         return 0;
53     }
54     int r = get_root();
```

```

55  if (n == 1) {
56      cout << 1;
57  } else if (n == 2) {
58      cout << "2 1";
59  } else if (n == 3) {
60      if (g[r].size() == 1) {
61          cout << "3 1 1";
62      } else {
63          cout << "2 2 1";
64      }
65  } else if (n == 4) {
66      solve4(r, 0);
67  } else if (n == 5) {
68      if (g[r].size() == 4) {
69          cout << "2 2 2 2 1";
70      } else if (g[r].size() == 3) {
71          cout << "3 2 2 1 1";
72      } else if (g[r].size() == 2) {
73          cout << "3 2 1 1 1";
74      } else if (g[r].size() == 1) {
75          r = g[r][0];
76          solve4(g[r][0], 1);
77      }
78  }
79  return 0;
80 }

```