Олимпиада «Ломоносов» по информатике. 2022-2023 учебный год
Работа участника с логином inf23f_247
Прайморадичная система счисления
Посылка по задаче 1 «Прайморадичная система счисления»

```
import sys

primes = [1, 1]
primorials = [1, 1]


def init():
    used = [False] * 10000
    for q in range(2, 10000):
        if used[q]:
            continue
        primes.append(q)
        for w in range(q * q, 10000, q):
            used[w] = True
    for q in range(2, 1000):
        primorials.append(primes[q] * primorials[q - 1])


def get_value(primorial):
    vec = primorial.split(":")
    vec = vec[::-1]
    result = 0
    for i in range(len(vec)):
        result += int(vec[i]) * primorials[i + 1]
    return result


init()

n = int(input())
a = []
values = []

for i in range(n):
    s = input()
    a.append(s)
    values.append(get_value(s))

if values.count(0) > 0:
```

```
    for i in range(n):
        if values[i] == 0:
            print(i + 1)
    sys.exit()

max_p = 0

for p in primorials:
    cnt = 0
    for val in values:
        if val % p == 0:
            cnt += 1
    if cnt > 0:
        max_p = max(max_p, p)

max_val = 0

for i in range(n):
    if values[i] % max_p == 0:
        max_val = max(values[i], max_val)

for i in range(n):
    if values[i] == max_val:
        print(i + 1)
```

Протокол проверяющей системы по задаче 1 «Прайморадичная система счисления»

```
OK
28 total tests runs, 28 passed, 0 failed.
Score gained: 100 (out of 100).
```

Посылка по задаче 2 «Сундуки»

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <string>
#include <map>

#define int long long

using namespace std;

signed main() {
    ios::sync_with_stdio(false);
```

```
cin.tie(0);
map<char, int> mp;
mp['a'] = 1;
mp['b'] = 5;
mp['c'] = 10;
mp['d'] = 50;
mp['e'] = 100;
mp['f'] = 200;
mp['g'] = 500;
mp['h'] = 1000;
mp['i'] = 2500;
mp['A'] = 500;
mp['B'] = 1000;
mp['C'] = 5000;
mp['D'] = 10000;
mp['E'] = 20000;
mp['F'] = 50000;
mp['G'] = 100000;
mp['H'] = 200000;
mp['I'] = 500000;
int n;
cin >> n;
vector<string> a(n);
vector<int> value(n);
for (int i = 0; i < n; i++) {
    cin >> a[i];
    for (int j = 0; j < (int) a[i].size(); j++) {
        value[i] += mp[a[i][j]];
    }
}
int max_diff = 0, max_dist = 0, K = -1, L = -1;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < i; j++) {
        int diff = abs(value[i] - value[j]);
        int dist = i - j;
        if (diff > max_diff || (diff == max_diff && dist > max_dist)) {
            max_diff = diff;
            max_dist = dist;
            K = j, L = i;
        }
    }
}
cout << K + 1 << '\n' << L + 1 << '\n';
return 0;
```

```
}
```

Протокол проверяющей системы по задаче 2 «Сундуки»

см. файл report2.txt

Посылка по задаче 3 «Кубик»

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <string>
#include <map>
#include <deque>

#define int long long

using namespace std;
map<string, string> dist;

const string finish = "1111223344556666";

void bfs(string start) {

    vector<string> moves;
    moves.push_back("L"), moves.push_back("R"), moves.push_back("U"), moves.push_back("D"), moves.push_back("D"), moves.push_back("F");
    dist[start] = "";
    deque<string> q;
    q.push_back(start);
    while (!q.empty()) {
        string v = q.front();
        q.pop_front();
        if (v == "1111223344556666") {
            cout << dist[finish];
            exit(0);
        }
        for (string move : moves) {
            if (move == "L") {
                string next(v);
                swap(next[12], next[0]);
                swap(next[14], next[2]);
                swap(next[8],  next[9]);
                swap(next[4], next[6]);
                if (dist.find(next) == dist.end()) {
```

```
        dist[next] = dist[v] + move;
        q.push_back(next);
    }
}
else if (move == "R") {
    string next(v);
    swap(next[1], next[13]);
    swap(next[3], next[15]);
    swap(next[10], next[11]);
    swap(next[5], next[7]);
    if (dist.find(next) == dist.end()) {
        dist[next] = dist[v] + move;
        q.push_back(next);
    }
}
else if (move == "U") {
    string next(v);
    swap(next[0], next[15]);
    swap(next[1], next[14]);
    swap(next[6], next[7]);
    swap(next[8], next[10]);
    if (dist.find(next) == dist.end()) {
        dist[next] = dist[v] + move;
        q.push_back(next);
    }
}
else if (move == "D") {
    string next(v);
    swap(next[2], next[13]);
    swap(next[3], next[12]);
    swap(next[4], next[5]);
    swap(next[9], next[11]);
    if (dist.find(next) == dist.end()) {
        dist[next] = dist[v] + move;
        q.push_back(next);
    }
}
else if (move == "F") {
    string next(v);
    next[0] = v[2];
    next[1] = v[0];
    next[2] = v[1];
    next[3] = v[3];
    next[4] = v[11];
```

```
                next[5] = v[10];
                next[6] = v[9];
                next[7] = v[8];
                next[8] = v[4];
                next[9] = v[5];
                next[10] = v[6];
                next[11] = v[7];
                next[12] = v[14];
                next[13] = v[12];
                next[14] = v[15];
                next[15] = v[13];

                if (dist.find(next) == dist.end()) {
                    dist[next] = dist[v] + move;
                    q.push_back(next);
                }
            }
        }
    }
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    string v;
    cin >> v;
    bfs(v);
    return 0;
}
```

Протокол проверяющей системы по задаче 3 «Кубик»

```
OK
50 total tests runs, 50 passed, 0 failed.
Score gained: 100 (out of 100).
```

Посылка по задаче 4 «Codemirror»

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <string>
#include <map>
#include <deque>
```

```
#define int int64_t

using namespace std;

string get_state(string &c, int time) {
    string text;
    string buf;
    int cursor = 0;
    int l = -1, r = -1;
    for (int i = 0; i < time; i++) {
        if (int(c[i]) >= 97 && c[i] <= 122) {
            text.insert(text.begin() + cursor, c[i]);
            cursor++;
            l = -1, r = -1;
        }
        else if (c[i] == '>') {
            cursor = min(cursor + 1, (int)text.size());
            l = -1, r = -1;
        }
        else if (c[i] == '<') {
            cursor = max(cursor - 1, (int) 0);
            l = -1, r = -1;
        }
        else if (c[i] == '{') {
            if (cursor == 0) {
                continue;
            }
            if (l == -1) {
                cursor--;
                l = cursor, r = cursor;
            }
            else {
                cursor--;
                l--;
            }
        }
        else if (c[i] == '}') {
            if (cursor == (int)text.size()) {
                continue;
            }
            if (l == -1) {
                l = cursor, r = cursor;
                cursor++;
```

```
    }
    else {
        cursor++;
        if (cursor == l + 1) {
            l++;
        }
        else {
            r++;

        }
    }
}
else if (c[i] == 'C') {
    buf.clear();
    if (l != -1) {
        for (int j = l; j <= r; j++) {
            buf.push_back(text[j]);
        }
    }
} else if (c[i] == 'V') {
    if (l != -1) {
        for (int j = l; j <= r; j++) {
            text.erase(text.begin() + l);
            cursor--;
        }
    }
    for (int j = 0; j < (int)buf.size(); j++) {
        text.insert(text.begin() + cursor + j, buf[j]);
    }
}
else if (c[i] == 'X') {
    buf.clear();
    if (l != -1) {
        for (int j = l; j <= r; j++) {
            if (cursor == l) {
                cursor--;
            }
            buf.push_back(*(text.begin() + l));
            text.erase(text.begin() + l);
        }
    }
}
else if (c[i] == 'D') {
    if (l != -1) {
```

```
                for (int j = l; j <= r; j++) {
                    text.erase(text.begin() + l);
                    if (cursor == r + 1) {
                        cursor--;

                    }
                }
            }
            else if (cursor != 0) {
                text.erase(text.begin() + cursor - 1);
                cursor--;
            }
        }

   }
    return text;
}

signed main() {
    int n;
    cin >> n;
    string c;
    cin >> c;
    int q;
    cin >> q;
    while (q--) {
        int time;
        cin >> time;
        cout << get_state(c, time) << endl;
        cout.flush();
    }
    return 0;
}
```

Протокол проверяющей системы по задаче 4 «Codemirror»

см. файл report4.txt

Посылка по задаче 5 «Библиотека»

```
import sys

queue_cnt = 0
used = dict()
owner = dict()
is_waiting = dict()
```

```
book_waiting = dict()
book_queue = dict()

lines = sys.stdin.readlines()
for line in lines:
    values = line.split()
    cmd = values[0]
    person = values[1]
    book = " ".join(values[2:])
    used[book] = False
    owner[book] = None
    is_waiting[person] = False
    book_queue[book] = []
    book_waiting[person] = None

max_queue_cnt = 0

for i in range(len(lines)):
    line = lines[i]
    values = line.split()
    cmd = values[0]
    person = values[1]
    book = " ".join(values[2:])
    if cmd == "B":
        if not used[book]:
            used[book] = True
            owner[book] = person
        else:
            if is_waiting[owner[book]] and owner[book_waiting[owner[book]]] == person:
                print(f"-{i + 1}")
                sys.exit()
            book_queue[book].append(person)
            is_waiting[person] = True
            book_waiting[person] = book
            queue_cnt += 1
    else:
        used[book] = False
        owner[book] = None
        if len(book_queue[book]) > 0:
            used[book] = True
            is_waiting[book_queue[book][0]] = True
            owner[book] = book_queue[book][0]
            book_queue[book] = book_queue[book][1:]
            book_waiting[person] = None
```

```
            queue_cnt -= 1
    max_queue_cnt = max(max_queue_cnt, queue_cnt)


print(max_queue_cnt)
```

Протокол проверяющей системы по задаче 5 «Библиотека»

см. файл report5.txt