

Олимпиада «Ломоносов» по информатике. 2022-2023 учебный год
Работа участника с логином inf23f_233
Прайморадичная система счисления
Посылка по задаче 1 «Прайморадичная система счисления»

p = []

```
def is_prime(n):
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d += 1
    return True

def from_prime_to_n(pr: str):
    data = []
    have_n = False
    for el in pr.split(':'):
        if not have_n and el == "0":
            continue
        have_n = True
        data.append(int(el))
    n = 0
    for i in range(len(data)):
        n += data[i] * p[len(data) - 1 - i]
    return n

def get_p():
    primes = [1]
    el = 2
    while len(primes) < 65:
        if is_prime(el):
            primes.append(el)
        el += 1
    for i in range(65):
        el = primes[0]
        for j in range(1, i + 1):
            el *= primes[j]
        p.append(el)
```

```

get_p()
data = []
n = int(input())
for _ in range(n):
    pr = input()
    data.append(from_prime_to_n(pr))
ans = []
for pr in p[::-1]:
    for i in range(n):
        if data[i] % pr == 0:
            ans.append(i)
    if ans:
        break
max_index = max(ans, key=lambda el: data[el])
for el in ans:
    if data[max_index] == data[el]:
        print(el + 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»

```

nominal = {
    "a": 0.01,
    "b": 0.05,
    "c": 0.1,
    "d": 0.5,
    "e": 1,
    "f": 2,
    "g": 5,
    "h": 10,
    "i": 25,
    "A": 5,
    "B": 10,
    "C": 50,
    "D": 100,
    "E": 200,
    "F": 500,
    "G": 1000,
    "H": 2000,
    "I": 5000
}

```

```

}

n = int(input())
data = []
for _ in range(n):
    s = input()
    val = 0
    for ch in s:
        val += nominal[ch]
    data.append(val)
max_diff = None
l_k_diff = None
k_ans = None
l_ans = None
for k in range(n):
    for l in range(k + 1, n):
        if max_diff is None or (abs(data[k] - data[l]) > max_diff and l - k > l_k_diff):
            k_ans = k
            l_ans = l
            max_diff = abs(data[k] - data[l])
            l_k_diff = l - k
print(k_ans + 1)
print(l_ans + 1)

```

Протокол проверяющей системы по задаче 2 «Сундуки»

см. файл report2.txt

Посылка по задаче 3 «Кубик»

```

#include <bits/stdc++.h>

using namespace std;

struct Element{
    vector<int> state;
    vector<char> p;
};

void do_l(vector<int>& arr){
    swap(arr[0],arr[12]);
    swap(arr[2],arr[14]);
    swap(arr[8],arr[9]);
    swap(arr[6],arr[4]);
}

```

```

void do_r(vector<int>& arr){
    swap(arr[1],arr[13]);
    swap(arr[3],arr[15]);
    swap(arr[10],arr[11]);
    swap(arr[5],arr[7]);
}
void do_u(vector<int>& arr){
    swap(arr[1],arr[14]);
    swap(arr[0],arr[15]);
    swap(arr[6],arr[7]);
    swap(arr[8],arr[10]);
}
void do_d(vector<int>& arr){
    swap(arr[3],arr[12]);
    swap(arr[2],arr[13]);
    swap(arr[9],arr[11]);
    swap(arr[4],arr[5]);
}
void do_f(vector<int>& arr){
    vector<int> new_v (16);
    new_v[0] = arr[2];
    new_v[1] = arr[0];
    new_v[2] = arr[3];
    new_v[3] = arr[1];
    new_v[4] = arr[11];
    new_v[5] = arr[10];
    new_v[6] = arr[9];
    new_v[7] = arr[8];
    new_v[8] = arr[4];
    new_v[9] = arr[5];
    new_v[10] = arr[6];
    new_v[11] = arr[7];
    new_v[12] = arr[13];
    new_v[13] = arr[15];
    new_v[14] = arr[12];
    new_v[15] = arr[14];
    for (int i=0; i<16;++i)
        arr[i] = new_v[i];
}
int main()
{
    ios::sync_with_stdio(0),cin.tie(0),cout.tie(0);
    vector<int> state(16);
    vector<int> need (16);

```

```

string s_need = "1111223344556666";
for (int i=0; i<16;++i)
    need[i] = s_need[i]-'0';
string s_state;
cin>>s_state;
for (int i=0;i<16;++i){
    state[i] = s_state[i]-'0';
}
queue<Element> q;
vector<char> e;
Element el = {state,e};
q.push(el);
bool do_while = true;
while (do_while){
    Element state_now = q.front();
    q.pop();
    bool equilar = true;
    for (int i=0;i<16;++i){
        if (state_now.state[i] != need[i]){
            equilar = false;
            break;
        }
    }
    if (equilar){
        for (int i=0; i<state_now.p.size();++i){
            cout<<state_now.p[i]<<' ';
        }
        do_while = false;
    }
    vector<int> do_l_v (16);
    vector<int> do_r_v (16);
    vector<int> do_u_v (16);
    vector<int> do_d_v (16);
    vector<int> do_f_v (16);
    vector<char> new_state_l;
    vector<char> new_state_r;
    vector<char> new_state_u;
    vector<char> new_state_d;
    vector<char> new_state_f;
    for (int i = 0; i<state_now.p.size();++i){
        new_state_l.push_back(state_now.p[i]);
        new_state_r.push_back(state_now.p[i]);
        new_state_u.push_back(state_now.p[i]);
        new_state_d.push_back(state_now.p[i]);
    }
}

```

```

        new_state_f.push_back(state_now.p[i]);
    }
    for (int i = 0; i<16;++i){
        do_l_v[i] = state_now.state[i];
        do_r_v[i] = state_now.state[i];
        do_u_v[i] = state_now.state[i];
        do_d_v[i] = state_now.state[i];
        do_f_v[i] = state_now.state[i];
    }
    do_l(do_l_v);
    new_state_l.push_back('L');
    Element l = {do_l_v,new_state_l};
    q.push(l);
    do_r(do_r_v);
    new_state_r.push_back('R');
    Element r = {do_r_v,new_state_r};
    q.push(r);
    do_u(do_u_v);
    new_state_u.push_back('U');
    Element u = {do_u_v,new_state_u};
    q.push(u);
    do_d(do_d_v);
    new_state_d.push_back('D');
    Element d = {do_d_v,new_state_d};
    q.push(d);
    do_f(do_f_v);
    new_state_f.push_back('F');
    Element f = {do_f_v,new_state_f};
    q.push(f);
}
}

```

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»

```

from sys import stdout

n = int(input())
s = input()
ans = [" " for _ in range(n + 1)]

```

```

s_now = ""
cur_now = 0
s_lite = [None, None]
buffer = None
for t in range(1, n + 1):
    command = s[t - 1]
    if command == "<":
        cur_now = max(0, cur_now - 1)
        s_lite = [None, None]
    elif command == ">":
        cur_now = min(len(s_now), cur_now + 1)
        s_lite = [None, None]
    elif command == "{":
        if cur_now > 0:
            cur_now -= 1
            if s_lite[0] is None:
                s_lite = [cur_now, cur_now]
            else:
                s_lite[0] -= 1
    elif command == "}":
        if cur_now <= len(s_now):
            if s_lite[0] is None:
                s_lite = [cur_now, cur_now]
            else:
                s_lite[1] += 1
                cur_now += 1
    elif command == "C":
        if s_lite[0] is None:
            buffer = None
        else:
            buffer = ''.join(s_now[i] for i in range(s_lite[0], s_lite[1] + 1))
    elif command == "V":
        if buffer is not None:
            if s_lite[0] is None:
                s_now = s_now[:cur_now] + buffer + s_now[cur_now:]
                cur_now += len(buffer)
            else:
                s_now = s_now[:s_lite[0]] + buffer + s_now[s_lite[1] + 1:]
                cur_now = cur_now - (s_lite[1] - s_lite[0]) + len(buffer) - 1
                s_lite = [None, None]
    elif command == "X":
        if s_lite[0] is not None:
            buffer = ''.join(s_now[i] for i in range(s_lite[0], s_lite[1] + 1))
            s_now = s_now[:s_lite[0]] + s_now[s_lite[1] + 1:]

```

```

        cur_now -= len(buffer)
    else:
        pass
elif command == "D":
    if s_lite[0] is None:
        s_now = s_now[:cur_now] + s_now[cur_now + 1:]
    else:
        s_now = s_now[:s_lite[0]] + s_now[s_lite[1] + 1:]
else:
    if s_lite[0] is not None:
        s_now = s_now[:s_lite[0]] + command + s_now[s_lite[1] + 1:]
        cur_now = s_lite[1] + 1 - (s_lite[1] - s_lite[0])
        s_lite = [None, None]
    else:
        s_now = s_now[:cur_now] + command + s_now[cur_now:]
        cur_now += 1
ans[t] = s_now
m = int(input())
for _ in range(m):
    req = int(input())
    print(ans[req])
    stdout.flush()

```

Протокол проверяющей системы по задаче 4 «Codemirror»

см. файл report4.txt

Посылка по задаче 5 «Библиотека»

```

from sys import stdin

person_books = {}
person_lock = {}
book_query = {}
book_person = {}
max_waiting = 0
# with open("data.txt") as file:
#     data = [el.split() for el in file.readlines()]
data = [el.split() for el in stdin.readlines()]
waiting_now = 0
for i in range(len(data)):
    req = data[i][0]
    name = data[i][1]
    book = ' '.join(data[i][2:])
    if name in person_lock and person_lock[name] == 1:
        continue

```



```

if req == "B":
    if book not in book_person:
        book_person[book] = name
        if name not in person_books:
            person_books[name] = {book}
        else:
            person_books[name].add(book)
    else:
        waiting_now += 1
        max_waiting = max(max_waiting, waiting_now)
        person_lock[name] = 1
        if book in book_person and book_person[book] in person_lock and person_lock[book_person[book]]:
            print(f"-{i + 1}")
            exit(0)
        if book in book_query:
            book_query[book].append(name)
        else:
            book_query[book] = [name]
else:
    del book_person[book]
    person_books[name].remove(book)
    if book in book_query and book_query[book]:
        max_waiting -= 1
        get_book_name = book_query[book][0]
        person_lock[get_book_name] = 0
        if get_book_name not in person_books:
            person_books[get_book_name] = {book}
        else:
            person_books[get_book_name].add(book)
        book_person[book] = get_book_name
        book_query[book] = book_query[book][1:]
print(max_waiting)

```

Протокол проверяющей системы по задаче 5 «Библиотека»

см. файл report5.txt