

Олимпиада «Ломоносов» по информатике. 2022-2023 учебный год

Работа участника с логином inf23f\_251

Прайморадичная система счисления

Посылка по задаче 1 «Прайморадичная система счисления»

```
def to_int(string):
    a = string.split(':')[::-1]
    res = 0
    for i in range(len(a)):
        res += arr[i] * int(a[i])
    return res
```

```
n = int(input())
max_ = -1
res = []
for k in range(n):
    q = to_int(input())
    if q == 0:
        if max_ is None:
            res.append((q, k))
        else:
            res = [(q, k)]
            max_ = None
    if max_ is None:
        continue
    if max_ == -1:
        max_ = 0
        res = [(q, k)]
    elif q % arr[max_] == 0:
        if q > res[-1][0]:
            res = [(q, k)]
        elif q == res[-1][0]:
            res.append((q, k))
    for i in range(max_ + 1, 65):
        if q % arr[i] == 0:
            max_ = i
            res = [(q, k)]
```

```
for i in res:
    print(i[1] + 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»

```
d = {
    'a': 1,
    'b': 5,
    'c': 10,
    'd': 50,
    'e': 100,
    'f': 2 * 100,
    'g': 5 * 100,
    'h': 10 * 100,
    'i': 25 * 100,
    'A': 5 * 100,
    'B': 10 * 100,
    'C': 50 * 100,
    'D': 100 * 100,
    'E': 200 * 100,
    'F': 500 * 100,
    'G': 1000 * 100,
    'H': 2000 * 100,
    'I': 5000 * 100,
}

def to_int(string):
    res = 0
    for i in string:
        res += d[i]
    return res

n = int(input())
min_ = None
max_ = None
mins = []
maxs = []

for i in range(n):
    q = to_int(input())
    if min_ is None or q < min_:
        min_ = q
        mins = [i]
    elif q == min_:
        mins.append(i)
    if max_ is None or q > max_:
        max_ = q
```

```

    maxs = [i]
elif q == max_:
    maxs.append(i)

if abs(mins[0] - maxs[-1]) > abs(mins[-1] - maxs[0]):
    if mins[0] < maxs[-1]:
        print(mins[0] + 1)
        print(maxs[-1] + 1)
    else:
        print(maxs[-1] + 1)
        print(mins[0] + 1)
else:
    if mins[-1] < maxs[0]:
        print(mins[-1] + 1)
        print(maxs[0] + 1)
    else:
        print(maxs[0] + 1)
        print(mins[-1] + 1)

```

### Протокол проверяющей системы по задаче 2 «Сундуки»

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

### Посылка по задаче 3 «Кубик»

```

from queue import Queue

nw = [0, 14, 6, 8] # fron, back, v, h
ne = [1, 15, 7, 10]
sw = [2, 12, 4, 9]
se = [3, 13, 5, 11]

def get_(state, idxs):
    return [state[idxs[0]], state[idxs[1]], state[idxs[2]], state[idxs[3]]]
def set_(state, idxs, vals):
    state[idxs[0]], state[idxs[1]], state[idxs[2]], state[idxs[3]] = vals[0], vals[1], vals[2], vals[3]

def replace(state, idxs1, idxs2):
    state[idxs1[0]], state[idxs2[0]], state[idxs1[1]], state[idxs2[1]] = state[idxs2[1]], state[idxs1[1]], state[idxs2[0]], state[idxs1[0]]
    state[idxs1[2]], state[idxs2[2]] = state[idxs2[2]], state[idxs1[2]]
    state[idxs1[3]], state[idxs2[3]] = state[idxs2[3]], state[idxs1[3]]

def rotate(state):

```

```

a, b, c, d = get_(state, nw), get_(state, ne), get_(state, sw), get_(state, se)
for x in [a, b, c, d]:
    x[2], x[3] = x[3], x[2]
set_(state, nw, c)
set_(state, ne, a)
set_(state, sw, d)
set_(state, se, b)

start_state = list(input())
q = Queue()
q.put((start_state, ''))

while True:
    state, way = q.get()

    if ''.join(state) == '1111223344556666':
        print(way)
        break

    new_state = state.copy()
    rotate(new_state)
    q.put((new_state, way + 'F'))

    new_state = state.copy()
    replace(new_state, nw, ne)
    q.put((new_state, way + 'U'))

    new_state = state.copy()
    replace(new_state, sw, se)
    q.put((new_state, way + 'D'))

    new_state = state.copy()
    replace(new_state, ne, se)
    q.put((new_state, way + 'R'))

    new_state = state.copy()
    replace(new_state, nw, sw)
    q.put((new_state, way + 'L'))

```

### 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»

```

import sys

_ = input()

text = ''
i = 0
start_select = None
end_select = None
last_select = False
buf = None

def has_selection():
    return start_select is not None

def unselect():
    global start_select, end_select
    start_select = None
    end_select = None

def insert(data):
    global text, i
    if not has_selection():
        if i == len(text):
            text = text[:i] + data
        else:
            text = text[:i] + data + text[i:]
        i += len(data)
    else:
        if end_select == len(text):
            text = text[:start_selected] + data
        else:
            text = text[:start_select] + data + text[end_select:]
        i = start_select + len(data)
        unselect()

def delete():
    global text, i
    if not has_selection():
        if i != 0:
            if i == len(text):
                text = text[:i - 1]

```

```

        else:
            text = text[:i - 1] + text[i:]
            i -= 1
    else:
        if end_select == len(text):
            text = text[:start_select]
        else:
            text = text[:start_select] + text[end_select:]
        i = start_select
        unselect()

res = []

for c in input():
    if has_selection() and start_select == end_select:
        unselect()
        last_select = False
    if c not in ['{', '}'] and last_select:
        last_select = False
        if end_select < start_select:
            start_select, end_select = end_select, start_select

    if ord(c) >= ord('a') and ord(c) <= ord('z'):
        insert(c)
    elif c == 'V':
        if buf is not None:
            insert(buf)

    elif c == 'C':
        if not has_selection():
            buf = None
        else:
            buf = text[start_select:end_select]
    elif c == 'D':
        delete()
    elif c == 'X':
        if not has_selection():
            pass
        else:
            buf = text[start_select:end_select]
            delete()

```

```

elif c == '<':
    if has_selection():
        unselect()
    if i > 0:
        i -= 1
elif c == '>':
    if has_selection():
        unselect()
    if i < len(text):
        i += 1

elif c == '{':
    if not last_select:
        last_select = True
        start_select = i
    if i > 0:
        i -= 1
    end_select = i
elif c == '}':
    if not last_select:
        last_select = True
        start_select = i
    if i < len(text):
        i += 1
    end_select = i

res.append(text)

for _ in range(int(input())):
    q = int(input())
    print(res[q])
    sys.stdout.flush()

```

**Протокол проверяющей системы по задаче 4 «Codemirror»**

см. файл report4.txt

**Посылка по задаче 5 «Библиотека» не было.**