Олимпиада «Ломоносов» по информатике. 2022-2023 учебный год
Работа участника с логином inf23f_269
Прайморадичная система счисления
Посылка по задаче 1 «Прайморадичная система счисления»

```
if 'Desktop' in __file__:
    input = open('input.txt').readline
else:
    input = open(0).readline


def from_p(s):
    s = s.split(':')[::-1]
    n = 0
    for i in range(len(s)):
        n += int(s[i]) * p[i]
    return n


p = [1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139,
 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311]
for i in range(1, len(p)):
    p[i] = p[i - 1] * p[i]
# print(p)

c = 0
smax = []
for i in range(1, int(input()) + 1):
    n = from_p(input().strip())
    # print(n)
    if n == 0:
        if c != -1:
            c = -1
            smax = []
        smax += [(n, i)]

    elif c != -1:
        if n % p[c] == 0:
            while c != -1 and n % p[c] == 0:
                c += 1
            smax = [(n, i)]
        elif n % p[c - 1] == 0:
            smax += [(n, i)]
```

```
nmax = max(smax)
for i in smax:
    if i[0] == nmax[0]:
        print(i[1])
```

Протокол проверяющей системы по задаче 1 «Прайморадичная система счисления»

см. файл report1.txt

Посылка по задаче 2 «Сундуки»

```
if 'Desktop' in __file__:
    input = open('input.txt').readline
else:
    input = open(0).readline


def from_s(s):
    n = 0
    for i in s:
        n += d[i]
    return n


d = {'a': 1, 'b': 5, 'c': 10, 'd': 50, 'e': 100, 'f': 200, 'g': 500, 'h': 1000, 'i': 2500, 'A': 500, 'B': 1000, 'C': 5000,
 'D': 10000, 'E': 20000, 'F': 50000, 'G': 100000, 'H': 200000, 'I': 500000}

nmax = (-float('inf'), 0, 0, 0)
p = [from_s(input().strip()) for _ in range(int(input()))]
# print(p)
for k in range(1, len(p) + 1):
    for l in range(k + 1, len(p) + 1):
        nmax = max(nmax, (abs(p[k - 1] - p[l - 1]), l - k, k, l))

# print(nmax)
print(*nmax[2:], sep='\n')
```

Протокол проверяющей системы по задаче 2 «Сундуки»

см. файл report2.txt

Посылка по задаче 3 «Кубик»

```
import sys

sys.setrecursionlimit(2000)
```

```
if 'Desktop' in __file__:
    input = open('input.txt').readline
else:
    input = open(0).readline


def L(s):
    s1 = list(s)
    s1[0], s1[12] = s1[12], s1[0]
    s1[2], s1[14] = s1[14], s1[2]
    s1[4], s1[6] = s1[6], s1[4]
    s1[8], s1[9] = s1[9], s1[8]
    return ''.join(s1)


def R(s):
    s1 = list(s)
    s1[1], s1[13] = s1[13], s1[1]
    s1[3], s1[15] = s1[15], s1[3]
    s1[5], s1[7] = s1[7], s1[5]
    s1[10], s1[11] = s1[11], s1[10]
    return ''.join(s1)


def U(s):
    s1 = list(s)
    s1[0], s1[15] = s1[15], s1[0]
    s1[1], s1[14] = s1[14], s1[1]
    s1[6], s1[7] = s1[7], s1[6]
    s1[8], s1[10] = s1[10], s1[8]
    return ''.join(s1)


def D(s):
    s1 = list(s)
    s1[2], s1[13] = s1[13], s1[2]
    s1[3], s1[12] = s1[12], s1[3]
    s1[4], s1[5] = s1[5], s1[4]
    s1[11], s1[9] = s1[9], s1[11]
    return ''.join(s1)


def F(s):
    s1 = list(s)
```

```
    s1[0] = s[2]
    s1[1] = s[0]
    s1[2] = s[3]
    s1[3] = s[1]
    s1[4] = s[11]
    s1[5] = s[10]
    s1[6] = s[9]
    s1[7] = s[8]
    s1[8] = s[4]
    s1[9] = s[5]
    s1[10] = s[6]
    s1[11] = s[7]
    s1[12] = s[13]
    s1[13] = s[15]
    s1[14] = s[12]
    s1[15] = s[14]
    return ''.join(s1)


def f(s):
    if s == '1111223344556666':
        return ''

    nmin = ' ' * 1001
    if s in d:
        if d[s]:
            return d[s]
        return nmin
    d[s] = ''

    for c, i in [('F', F), ('L', L), ('R', R), ('U', U), ('D', D)]:
        try:
            nmin = min(nmin, c + f(i(s)), key=len)
        except:
            pass
    # print(nmin)
    d[s] = nmin
    return nmin


# s = input().strip()
# print(F('1111223344556666'))
d = {}
print(f(input().strip()))
```

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»

```
def main():
    from math import isqrt

    if 'Desktop' in __file__:
        input = open('input.txt').readline
    else:
        input = open(0).readline
    out = open(1, 'w')


    def eva(commands, start=('', 0, None, '', 0)):
        s, cur, hl, buf, shift = start
        for i in commands:
            if i == '>':
                if hl and not shift:
                    cur = max(hl)
                    hl = None
                else:
                    if cur < len(s) - 1:
                        cur += 1
                    if shift:
                        if hl:
                            hl = hl[0], cur
                        else:
                            hl = cur - 1, cur

            elif i == '<':
                if hl and not shift:
                    cur = min(hl)
                    hl = None
                else:
                    if cur > 0:
                        cur -= 1
                    if shift:
                        if hl:
                            hl = hl[0], cur
                        else:
```

```
                hl = cur + 1, cur

elif i == '{':
    shift = True
elif i == '}':
    shift = False

elif i == 'C':
    if hl:
        buf = s[min(hl):max(hl)]
    else:
        buf = ''

elif i == 'V':
    if hl:
        cur -= max(0, hl[1] - hl[0])
        s = s[:min(hl)] + s[max(hl):]
        if not shift:
            hl = None
    s = s[:cur] + buf + s[cur:]
    cur += len(buf)
    if hl:
        hl = (cur, cur)

elif i == 'X':
    if hl:
        buf = s[min(hl):max(hl)]
        cur -= max(0, hl[1] - hl[0])
        s = s[:min(hl)] + s[max(hl):]
        hl = (cur, cur)
        if not shift:
            hl = None
    else:
        buf = ''

elif i == 'D':
    if hl:
        cur -= max(0, hl[1] - hl[0])
        s = s[:min(hl)] + s[max(hl):]
        hl = (cur, cur)
        if not shift:
            hl = None
    elif cur > 0:
        s = s[:cur - 1] + s[cur:]
```

```
                cur -= 1

            else:
                s = s[:cur] + i + s[cur:]
                cur += 1
        return s, cur, hl, buf, shift


    _ = int(input())
    commands = input().strip()
    m = isqrt(len(commands))
    # m = 4

    s = ''
    cur = 0
    hl = None
    buf = ''
    shift = 0
    p = [(s, cur, hl, buf, shift)]
    for i in range(len(commands) // m):
        s, cur, hl, buf, shift = eva(commands[i * m:i * m + m], (s, cur, hl, buf, shift))
        p += [(s, cur, hl, buf, shift)]
        # print(commands[i * m:i * m + m], '-', '"' + s[:cur] + '|' + s[cur:] + '"', hl, buf, shift)

    for t in range(int(input())):
        n = int(input())
        s, *_ = eva(commands[n // m * m:n], p[min(n // m, len(p) - 1)])
        out.write(s + '\n')
        out.flush()


main()
```

Протокол проверяющей системы по задаче 4 «Codemirror»

см. файл report4.txt

Посылка по задаче 5 «Библиотека»

```
from collections import deque

books = {}
nmax = 0
for n, i in enumerate(open('input.txt')):
    c, reader, *name = i.strip().split(' ')
    name = ' '.join(name)
```

```
    if c == 'B':
        if name not in books:
            books[name] = deque()
        others = {i: books[name].index(i) for i in books[name]}
        for book in books:
            for other in min((books[book], others), key=len):
                if reader in books[book] and other in books[book]:
                    if books[book].index(other) > books[book].index(reader):
                        print(-n - 1)
                        exit(0)
        books[name].append(reader)
        nmax = max(nmax, len(books[name]))

    else:
        books[name].popleft()
print(nmax - 1)
```

Протокол проверяющей системы по задаче 5 «Библиотека»

см. файл report5.txt