

Федеральное государственное автономное образовательное  
учреждение высшего образования  
Университет ИТМО

Факультет инфокоммуникационных технологий

## **Алгоритмы и структуры данных:**

**Отчёт по лабораторной работе №6: Деревья.  
Пирмида, пирамидальная сортировка. Очередь  
с приоритетами.**

Выполнил:  
**Бочкарь Артём Артёмович**

Группа: **K32392**

Преподаватели:  
**Артамонова В. Е.**

Санкт-Петербург 2023 г.

## Задача №1: Множество

В первом задании нужно было реализовать множество с операциями “добавление ключа”, “удаление ключа” и “проверка существования ключа”. Надо последовательно вывести результаты выполнения всех операций “?”:

```
my_set = set()
input = open('input.txt', 'r')
output = open('output.txt', 'w')
k = 0
n = int(input.readline())
while k != n:
    a = input.readline()
    a = a.split()
    if a[0] == 'A':
        my_set.add(a[1])
    if a[0] == 'D':
        my_set.discard(a[1])
    if a[0] == '?':
        if a[1] in my_set:
            output.write('Y' + '\n')
        else:
            output.write('N' + '\n')
    k += 1
```

Тесты:

input.txt		output.txt	
1	8	1	Y
2	A 2	2	N
3	A 5	3	N
4	A 3	4	
5	? 2		
6	? 4		
7	A 2		
8	D 2		
9	? 2		

## Задача №2: Телефонная книга

Во втором задании надо было реализовать простой менеджер телефонной книги, который сможет обрабатывать следующие команды: “add number name”, “del number”, “find number”, а вывести надо результат каждого поискового запроса find:

```
numbers = {}
input = open('input.txt', 'r')
output = open('output.txt', 'w')
k = 0
n = int(input.readline())
while k != n:
    a = input.readline()
    a = a.split()
    if len(a) == 0:
        k += 1
        continue
    if a[0] == 'add':
        numbers[a[1]] = a[2]
    if a[0] == 'find':
        if a[1] in numbers:
            output.write(numbers[a[1]] + '\n')
        else:
            output.write("not found" + '\n')
    if a[0] == 'del':
        if a[1] in numbers:
            del numbers[a[1]]
        else:
            continue
    k += 1
```

Тесты:

input.txt	output.txt
1 8	1 not found
2 find 3839442	2 granny
3 add 123456 me	3 me
4 add 0 granny	4 not found
5 find 0	5
6 find 123456	
7 del 0	
8 del 0	
9 find 0	

## Задача №5: Выборы в США

В пятом задании нужно было подвести итоги выборов в США, подсчитав количество отданных голосов за каждого из кандидатов. Каждая строка входного файла содержит фамилию кандидата, за которого отдают голоса выборщики этого штата, затем через пробел идет количество выборщиков, отдавших голоса за этого кандидата. Нужно вывести фамилии всех кандидатов в лексикографическом порядке, затем, через пробел, количество отданных за них голосов:

```
presidents = {}
input_txt = open('input.txt', 'r')
output_txt = open('output.txt', 'w')
n = int(input())
k = 0
while k != n:
    a = input_txt.readline()
    a = a.split()
    if a[0] in presidents:
        presidents[a[0]] += int(a[1])
    else:
        presidents[a[0]] = int(a[1])
    k += 1

for i in sorted(presidents):
    output_txt.write(i + ' ')
    output_txt.write(str(presidents[i]) + '\n')
```

Тесты:

input.txt			output.txt	
1	McCain 10	✓	1	McCain 16
2	McCain 5		2	Obama 17
3	Obama 9		3	
4	Obama 8			
5	McCain 1			

## Задача №6: Фибоначчи возвращается

В 6 задачи даётся последовательность чисел и требовалось для каждого числа определить является ли оно числом Фибоначчи. Для каждого числа, данного во входном файле, выведите «Yes», если оно является числом Фибоначчи, и «No» в противном случае:

```
input_txt = open('input.txt')
output_txt = open('output.txt', 'w')
n = input_txt.readline()
n = int(n)
fib = list()
f = 0
for i in range(0, n):
    j = int(input_txt.readline())
    fib.append(j)
    f = max(f, j)
q, w = 1, 1
sed = set()
sed.add(1)
while w <= f:
    c = q + w
    sed.add(c)
    q = w
    w = c
for j in fib:
    if j in sed:
        output_txt.write('Yes\n')
    else:
        output_txt.write('No\n')
input_txt.close()
output_txt.close()
```

Тесты:

## Задача №7: Драгоценные камни

В этой задаче надо было найти пары драгоценных камней, которые нашёл визирь, чтобы шах приобрёл найденные красивые пары. Первая строка входного файла содержит целые числа  $n$  и  $k$  ( $1 \leq n \leq 100000$ ,  $1 \leq k \leq 676$ ) – число камней, которые привезли купцы и число пар, которые шах считает красивыми. Вторая строка входного файла содержит строку  $S$ , описывающую типы камней, которые привезли купцы. Далее следуют  $k$  строк, каждая из которых содержит две строчных буквы латинского алфавита и описывает одну из красивых пар камней. В выходной файл нужно вывести ответ на задачу – количество пар, которое должен найти визирь:

```
input_txt = open("input.txt")
output_txt = open("output.txt", "w")
n, k = map(int, input_txt.readline().split())
f = input_txt.readline()

dict = {}
for i in range(0, 26):
    st = chr(i + 97)
    dict[st] = list()
    if f[0] != st:
        dict[st].append(0)
    else:
        dict[st].append(1)
    for j in range(1, n):
        dict[st].append(dict[st][-1])
        if f[j] == st:
            dict[st][-1] += 1

ans = 0
for i in range(0, k):
    kamny = input_txt.readline()
    for j in range(0, n - 1):
        if f[j] == kamny[0]:
            ans += dict[kamny[1]][n - 1] - dict[kamny[1]][j]

output_txt.write(str(ans))
```

Тесты:

input.txt		output.txt	
1	7 1	1	6
2	abacaba		
3	aa		