

# Последовательности

## Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Аргументы командной строки идут в порядке аргументов – входных данных задачи.
3. Основная программа (`main`) должна вводить аргументы (с командной строки), вызывать функцию, реализующую задачу, и выводить результат(ы) ее работы.

## Задачи

1. Написать функцию, получающую в качестве аргумента имя файла, содержащего последовательность битов неизвестной длины (в файле битовая последовательность задается как последовательность целых чисел, двоичное представление которых (без ведущих 0) образует битовую последовательность), а также целое число  $k$ , и возвращающую целое число, равное минимальному расстоянию между соседними участками этой последовательности, составленными из единичных битов и имеющими длину не менее  $k$ . Функция должна возвращать  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл, прочитать элемент и т.д..
2. Написать функцию, получающую в качестве аргументов имена двух файлов, содержащих последовательности  $a_1, a_2, \dots$  и  $b_1, b_2, \dots$  вещественных чисел неизвестной длины. Функция должна возвращать целое число, равное 1, если каждый элемент второй последовательности (кроме первого и последнего) равен полусумме элементов первой последовательности с соседними ему номерами:  $b_i = (a_{i+1} + a_{i-1})/2$ , и 0 в противном случае. Функция должна возвращать  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл, прочитать элемент и т.д..
3. Написать функцию, получающую в качестве аргумента имя файла, содержащего последовательность вещественных чисел неизвестной длины, и возвращающую целое число, равное 1, если эта последовательность является арифметической прогрессией, 2, если она является геометрической прогрессией, 3, если постоянна, 4, если в ней недостаточно элементов для принятия решения, и 0 в противном случае. Функция должна возвращать  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл, прочитать элемент и т.д..
4. Написать функцию, получающую в качестве аргументов имя файла, содержащего последовательность вещественных чисел неизвестной длины, и адрес вещественного числа  $d$ , и возвращающую в  $d$  вещественное число, равное среднему квадратическому отклонению чисел из этого файла от их среднего арифметического. Возвращаемое значение функции равно длине последовательности в случае успешного завершения и равно  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл, прочитать элемент и т.д., при этом переменная  $d$  не меняется.

# Массивы

## Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Аргументы командной строки для задач 1–3:
  - 1)  $X$  – дополнительный аргумент:  $X = k$  в задачах 1 и 2 ( $k$  – целое число),  $X = b$  в задаче 3 ( $b$  – имя файла (текстовая строка)),
  - 2)  $n$  – размерность массива,

- 3)  $p$  – количество выводимых значений в массиве,
- 4)  $s$  – задает номер формулы для инициализации массива, должен быть равен 0 при вводе массива из файла,
- 5) `filename` – имя файла, откуда надо прочитать массив. Этот аргумент **отсутствует**, если  $s! = 0$ .

### 3. Аргументы командной строки для задачи 4:

- 1)  $n$  – размерность массива  $a$ ,
- 2)  $p_a$  – количество выводимых значений в массиве  $a$ ,
- 3)  $s_a$  – задает номер формулы для инициализации массива  $a$ , должен быть равен 0 при вводе массива из файла,
- 4) `filenamea` – имя файла, откуда надо прочитать массив. Этот аргумент **отсутствует**, если  $s_a! = 0$ ,
- 5)  $m$  – размерность массива  $b$ ,
- 6)  $p_b$  – количество выводимых значений в массиве  $b$ ,
- 7)  $s_b$  – задает номер формулы для инициализации массива  $b$ , должен быть равен 0 при вводе массива из файла,
- 8) `filenameb` – имя файла, откуда надо прочитать массив. Этот аргумент **отсутствует**, если  $s_b! = 0$ .

4. Ввод массива должен быть оформлен в виде подпрограммы.

5. Ввод массива из файла. В указанном файле находится массив в формате:

```
число-1  число-2
...
число-n
```

где  $n$  - указанный размер массива. Числа разделяются либо пробелами, либо символом новой строки. Программа должна выводить сообщение об ошибке, если указанный файл не может быть прочитан, содержит меньшее количество данных или данные неверного формата.

6. Ввод массива по формуле. Элемент  $a_i$  массива  $A$  полагается равным

$$a_i = f(s, n, i), \quad i = 1, \dots, n,$$

где  $f(s, n, i)$  - функция, которая возвращает значение  $(i)$ -го элемента массива по формуле номер  $s$  (аргумент командной строки). Функция  $f(s, n, i)$  должна быть оформлена в виде отдельной подпрограммы.

$$f(s, n, i) = \begin{cases} i & \text{при } s = 1 \\ n - i & \text{при } s = 2 \\ i/2 & \text{при } s = 3 \\ n - i/2 & \text{при } s = 4 \\ 2i & \text{при } s = 5 \\ n - 2i & \text{при } s = 6 \end{cases}$$

7. Решение задачи должно быть оформлено в виде подпрограммы, получающей в качестве аргументов массив и его длину (в ряде задач также дополнительные аргументы). Получать в этой подпрограмме дополнительную информацию извне т.п. запрещается.

8. Программа должна содержать подпрограмму вывода на экран массива длины не более  $p$ . Эта подпрограмма используется для вывода исходного массива после его инициализации, а также для вывода на экран результата. Подпрограмма выводит на экран не более, чем  $p$  элементов массива, где  $p$  – параметр этой подпрограммы (аргумент командной строки). Каждый элемент массива должен печататься на новой строке.
9. Программа должна выводить на экран время, затраченное на решение.

### Задачи

1. Написать подпрограмму, получающую в качестве аргументов целое число  $k$ , массив вещественных чисел, целое число  $n$ , являющееся длиной этого массива, и циклически сдвигающую элементы массива на  $k$  позиций вправо произведя не более  $n + O(1)$  перемещений элементов. Основная программа должна заполнять данными массив, выводить его на экран, вызывать эту подпрограмму и выводить на экран результат ее работы.
2. Написать функцию, получающую в качестве аргументов целое число  $k$ , массив  $a[n]$  вещественных чисел, целое число  $n$ , являющееся длиной этого массива, и выбрасывающую из массива все элементы, меньшие среднего арифметического элементов исходного массива, находящихся в участках невозрастания исходного массива длиной не менее  $k$  (при выбрасывании элемента  $a[i]$  все элементы с номером, большим  $i$ , сдвигаются на одну позицию к началу массива и длина массива уменьшается на 1), произведя при этом не более  $2n + O(1)$  пересылок элементов. Функция возвращает длину получившегося в результате массива. Основная программа должна заполнять данными массив, выводить его на экран, вызывать эту функцию и выводить на экран результат ее работы.
3. Написать функцию, получающую в качестве аргументов имя файла  $b$ , массив  $a[n]$  вещественных чисел, целое число  $n$ , являющееся длиной этого массива, и возвращающую целое число, равное количеству вхождений подпоследовательности  $a[n]$  в последовательность, содержащуюся в заданном файле  $b$ . Под вхождением понимается совпадение  $n$  подряд идущих элементов последовательности в файле  $b$  с элементами массива  $a[0], \dots, a[n-1]$ . Например, для массива  $a[2] = \{1, 1\}$  и последовательности  $b = \{1, 1, 1, 1, 1\}$  число вхождений равно 4. Функция должна возвращать  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл  $b$ , прочитав элемент и т.д.. Основная программа должна заполнять данными массив, выводить его на экран, вызывать эту функцию и выводить на экран результат ее работы.
4. Написать функцию, получающую в качестве аргументов три массива  $a[n]$ ,  $b[m]$ ,  $c[n+m]$  вещественных чисел, где  $a[n]$ ,  $b[m]$  строго возрастают, и целые числа  $n$  и  $m$ , и строящую по строго возрастающим массивам  $a[n]$ ,  $b[m]$  строго возрастающий массив  $c[n+m]$  слиянием первых двух (с выбрасыванием повторяющихся чисел) за  $n + m + O(1)$  сравнений и  $n + m + O(1)$  пересылок элементов по следующему алгоритму

Просматриваем очередные элементы  $a[i]$ ,  $i = 0, \dots, n-1$  и  $b[j]$ ,  $j = 0, \dots, m-1$  массивов  $a$  и  $b$ . Если  $a[i] < b[j]$ , то  $c[k] = a[i]$  и увеличиваем  $i$  на 1, если  $a[i] > b[j]$ , то  $c[k] = b[j]$  и увеличиваем  $j$  на 1, иначе увеличиваем на 1  $i$  или  $j$ . Здесь  $k$ ,  $0 \leq k \leq n+m-1$  – очередной элемент массива  $c$ .

Функция должна возвращать длину получившегося в результате массива  $c$ . Основная программа должна заполнять данными массивы  $a$  и  $b$ , выводить их на экран, вызывать эту подпрограмму и выводить на экран результат ее работы – массив  $c$ .