

Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Задачи оцениваются независимо в двух группах: задачи 1–4 и задачи 5–8.
3. Аргументы командной строки для задач 1–8:

1) name – имя файла.

Например, запуск

```
./a.out a.txt
```

означает, что имя файла – это `a.txt`.

4. Решение задачи должно быть оформлено в виде функции.
5. В функции, реализующей решение задачи, запрещается что-либо выводить на экран.
6. В программе должна быть функция

```
int print_bits (const char*);
```

получающая 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), и выводящая эту битовую последовательность на экран в одну строчку и без пробелов. Функция должна возвращать `-1`, `-2` и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д..

7. В программе должна быть функция

```
int print_bits0 (const char*);
```

получающая в качестве аргумента имя файла, содержащего последовательность битов неизвестной длины (в файле битовая последовательность задается как последовательность целых чисел, двоичное представление которых (с ведущими 0) образует битовую последовательность), и выводящая эту битовую последовательность на экран в одну строчку и без пробелов. Функция должна возвращать `-1`, `-2` и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д..

8. Функции `print_bits` и `print_bits0` размещаются в отдельном файле с именем `print_bits.c`, а их прототипы – в файле `print_bits.h`. Файл `print_bits.h` включается в файл `print_bits.c` и в файлы, где находятся реализации заданий.
9. Пример файла `print_bits.h`:

```
int print_bits (const char*);  
int print_bits0 (const char*);
```

10. Пример файла `print_bits.c`:

```
#include <stdio.h>
#include "io_status.h"
#include "print_bits.h"
/* реализация функций */
...
```

11. Пример файла реализующей задание:

```
#include <stdio.h>
#include "io_status.h"
#include "print_bits.h"
/* реализация функции, решающей задачу, и функции main */
...
```

12. Файл `print_bits.c` добавляется в команду компиляции задач. Пример компиляции для задачи 1:

```
gcc task01.c print_bits.c -o a01.out
```

13. Вывод результата работы в функции `main` для задач **1, 3, 5, 7** должен производиться по формату:

```
print_bits (argv[1]);
printf ("%s : Task = %d Result = %d i = %d j = %d\n",
        argv[0], task, res, i, j);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,
- `res` – возвращаемое значение функции, реализующей решение этой задачи,
- `i, j` – значения переменных `i, j` после работы функции, реализующей решение этой задачи.

14. Вывод результата работы в функции `main` для задач **2, 4, 6, 8** должен производиться по формату:

```
print_bits0 (argv[1]);
printf ("%s : Task = %d Result = %d i = %d j = %d\n",
        argv[0], task, res, i, j);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,
- `res` – возвращаемое значение функции, реализующей решение этой задачи,
- `i, j` – значения переменных `i, j` после работы функции, реализующей решение этой задачи.

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 адреса целых переменных i , j , и возвращающую целое число, равное количеству элементов этой последовательности, больших предыдущего элемента. Целая переменная i получает значение, равное номеру первого такого бита в последовательности, переменная j получает значение, равное номеру последнего такого бита в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
2. 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 (с ведущими 0) образует битовую последовательность), а также адреса целых переменных i , j , и возвращающую целое число, равное количеству элементов этой последовательности, больших двух следующих элементов. Целая переменная i получает значение, равное номеру первого такого бита в последовательности, переменная j получает значение, равное номеру последнего такого бита в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
3. 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 адреса целых переменных i , j , и возвращающую целое число, равное количеству строгих локальных минимумов этой последовательности (т.е. количеству битов x_i таких, что $x_{i-1} > x_i < x_{i+1}$). Целая переменная i получает значение, равное номеру первого локального минимума в последовательности, переменная j получает значение, равное номеру последнего локального минимума в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
4. 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 (с ведущими 0) образует битовую последовательность), а также адреса целых переменных i , j , и возвращающую целое число, равное количеству строгих локальных максимумов этой последовательности (т.е. количеству битов x_i таких, что $x_{i-1} < x_i > x_{i+1}$). Целая переменная i получает значение, равное номеру первого локального максимума в последовательности, переменная j получает значение, равное номеру последнего локального максимума в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
5. 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 адреса целых переменных i , j , и возвращающую целое число, равное длине максимального участка, составленного из единичных битов, этой последовательности. Целая переменная i получает значение, равное номеру бита, с которого начинается первый такой участок в последовательности, переменная j получает значение, равное номеру бита, с которого начинается последний такой участок в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.

6. 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 (с ведущими 0) образует битовую последовательность), а также адреса целых переменных i , j , и возвращающую целое число, равное длине максимального участка, составленного из нулевых битов, этой последовательности. Целая переменная i получает значение, равное номеру бита, с которого начинается первый такой участок в последовательности, переменная j получает значение, равное номеру бита, с которого начинается последний такой участок в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
7. 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 адреса целых переменных i , j , и возвращающую целое число, равное количеству участков, составленного из единичных битов, этой последовательности. Целая переменная i получает значение, равное номеру бита, с которого начинается первый такой участок в последовательности, переменная j получает значение, равное номеру бита, с которого начинается последний такой участок в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.
8. 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 (с ведущими 0) образует битовую последовательность), а также адреса целых переменных i , j , и возвращающую целое число, равное количеству участков, составленных из нулевых битов, этой последовательности. Целая переменная i получает значение, равное номеру бита, с которого начинается первый такой участок в последовательности, переменная j получает значение, равное номеру бита, с которого начинается последний такой участок в последовательности. Нумерация битов начинается с 0. Функция должна возвращать -1 , -2 и т.д. (как именованную константу, описанную в файле `io_status.h`), если она не смогла открыть файл, прочитать элемент и т.д., при этом переменные i , j не изменяются.