Problem mz04-1-mz04-1

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/26 10:30:00
Deadline:	2022/09/26 12:05:00

Problem mz04-1: mz04-1

Компьютерная система Pensieve имеет следующие свойства: CHAR_BIT == 12, sizeof(unsigned short) == 2, целые числа записываются в порядке байт Big endian. Для совместимости с компьютерами, у которых CHAR_BIT == 8, файлы для Pensieve записываются в формате, в котором каждые 12 бит хранятся в двух 8-битных байтах в Big endian формате. В первом (старшем) байте используются только 4 младших бита, а старшие 4 бита всегда нулевые.

Таким образом, 24-битное число 0x123456 будет храниться в файле для обработки на Pensieve как последовательность байт 0x01 0x23 0x04 0x56.

Напишите программу, которая будет формировать выходной файл для Pensieve из последовательности чисел в десятичном виде. На стандартном потоке ввода подается последовательность 24-битных беззнаковых целых чисел в десятичном виде. Имя выходного файла указывается в командной строке. Выходной файл должен быть создан с правами только на чтение и запись только для владельца, если он не существовал. Если файл существовал, права доступа не меняйте.

Для записи выходного файла используйте POSIX API. Для записи чисел используйте системный вызов `write`, в который передается буфер размера 32 бита.

Предполагайте, что на компьютере, на котором выполняется программа, CHAR_BIT == 8. Все системные вызовы завершаются успешно.

Problem mz04-2-mz04-2

Full score:	100
Run penalty:	10
Input file name:	input
Output file name:	output
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/26 10:30:00
Deadline:	2022/09/26 12:05:00

Problem mz04-2: mz04-2

Программе передаются два аргумента командной строки: имя файла для обработки (FILE), затем число элементов для обработки (N). Файл для обработки — это бинарный файл, содержащий вещественные числа двойной точности. Размер файла всегда кратен размеру одного числа. Если файл не содержит чисел или содержит только одно число, он не изменяется. Если файл содержит более одного числа, первое число не изменяется, а i-е число в файле после обработки вычисляется по формуле out[i] = in[i] - out[i - 1], где out – содержимое файла после обработки, а in – содержимое файла до обработки. Число N определяет максимальное количество обрабатываемых чисел, то есть если в файле более N чисел, то обрабатываются первые N чисел, а остальные не изменяются. in[0] учитывается в числе N. Таким образом, при N < 1 входной файл вообще не изменяется.

Для работы с файлом необходимо использовать средства низкоуровневого ввода-вывода. Программа должна завершать свое выполнение с кодом возврата 0. Не держите в памяти больше двух чисел из файла одновременно. Для перемещения по файлу используйте `lseek`.

Пример входных данных: 1 2 3 4 5 6 7 8 9

Результат работы при N = 4: 1 1 2 2 5 6 7 8 9

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
913	2022/09/26 11:57:43	688	mz04-2	gcc-32	Partial solution	0	**0**=0-1*10	View	View
903	2022/09/26 11:41:13	1013	mz04-2	gcc-32	Partial solution	0	**0**	View	View
901	2022/09/26 11:40:39	1023	mz04-2	gcc-32	Compilation error	N/A	N/A	View	View

Problem mz04-3-mz04-3

Full score:	100
Run penalty:	10
Input file name:	input.txt
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/26 10:30:00
Deadline:	2022/09/26 12:05:00

Problem mz04-3: mz04-3

В аргументах командной строки задается имя входного бинарного файла произвольного доступа. Бинарный файл рассматривается как массив структур:

```
struct Node
{
    int32_t key;
    int32_t left_idx;
    int32_t right_idx;
};
```

Структура описывает вершину бинарного дерева поиска, где `key` – ключ поиска, а значения `left_idx` и `right_idx` – индексы в массиве соответственно левой и правой вершины. Корень дерева находится в элементе массива с индексом 0. Признаком отсутствия соответствующего поддерева в вершине является индекс, равный 0.

Числа в файле хранятся в естественном для x86 представлении (little-endian).

На стандартный поток вывода напечатайте ключи, размещенные в данном дереве, в порядке убывания.

Считывать содержимое всего файла в память целиком одним блоком запрещается. Для чтения содержимого файла необходимо пользоваться низкоуровневым вводом-выводом (open/read/etc). Корректная работа на big-endian архитектуре не требуется. Не используйте низкоуровневые манипуляции с памятью (union, приведение типов указателей). Не используйте библиотечные функции преобразования.

Для вывода ключей можно использовать высокоуровневые потоки (stdout).

Не забывайте выводить '\n' в конце вывода.

В программе должна быть реализована проверка корректности чтения из файла. Минимизируйте число системных вызовов, необходимых для чтения одной записи из файла.

mz01-1
mz01-2
mz01-3
mz01-4
mz01-5
mz01-6
up01-1
up01-2
up01-3
up01-4
up01-5
up01-6
mz02-1
mz02-2
mz02-3
mz02-4
mz02-5
mz02-6
up02-1
up02-2
up02-3
up02-4
up02-5
up02-6
up02-7
mz03-1
mz03-2
mz03-3
mz03-4
mz03-5
up03-1
up03-2
up03-3
up03-4
up03-5
mz04-1
mz04-2
mz04-3
mz04-4
mz04-5
mz04-6
up04-1
up04-2
up04-3
up04-4
up04-5
mz05-1
mz05-2
mz05-3
mz05-4
mz05-5
up05-1
up05-2
up05-3
up05-4
up05-5
mz06-1
mz06-2
mz06-3
mz06-4
mz06-5
up06-1
up06-2
up06-3
up06-4
up06-5
mz07-1
mz07-2
mz07-3
mz07-4
mz07-5
up07-1
up07-2
up07-3
up07-4
up07-5
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5

Михайленко Ярослав [204 группа, 3 семестр, 2022-2023]: Submit a solution

Submit a solution for up04-4-mz04-4 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M

Problem up04-4: mz04-4 (дореш)

В аргументах командной строки задается путь к бинарному файлу, содержащему целые знаковые числа типа long long, записанные в порядке байт хоста (компьютера, на котором исполняется программа). В файле найдите минимальное число и поменяйте его знак. Если минимальное число в файле встречается несколько раз, поменяйте знак только у первого по порядку минимального числа. При вычислениях предполагайте неограниченную битность, но в качестве результата оставьте младшие 64 бита.

Для работы с файлом используйте POSIX API. Допускается только один проход по файлу, но бинарный файл является файлом произвольного доступа.

На 32-битных системах компилируйте вашу программу с опцией `-D_FILE_OFFSET_BITS=64`. Не допускайте в программе undefined behavior.

Например, если дан входной бинарный файл input.bin, результатом обработки его должен быть файл распечатка содержимого которого показана ниже.

```
hexdump -C input.bin
00000000  03 00 00 00 00 00 00 00  fe ff ff ff ff ff ff ff  |................|
00000010  04 00 00 00 00 00 00 00                           |........|
00000018
```

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File: Выберите файл | Файл не выбран

Send! Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
1477	2022/10/06 16:31:01	1278	up04-4	gcc-32	Pending review	9	**50**=100-5*10	View	View
1476	2022/10/06 16:27:27	1257	up04-4	gcc-32	Partial solution	1	**0**=0-4*10	View	View
1474	2022/10/06 16:12:10	1239	up04-4	gcc-32	Partial solution	1	**0**=0-3*10	View	View
1473	2022/10/06 16:11:52	1239	up04-4	gcc-32	Compilation error	N/A	N/A	View	View
1470	2022/10/06 15:58:27	1231	up04-4	gcc-32	Partial solution	1	**0**=0-2*10	View	View
1452	2022/10/06 08:59:00	1099	up04-4	gcc-32	Partial solution	1	**0**=0-1*10	View	View
1451	2022/10/06 08:41:35	1093	up04-4	gcc-32	Partial solution	1	**0**	View	View

mz01-1
mz01-2
mz01-3
mz01-4
mz01-5
mz01-6
up01-1
up01-2
up01-3
up01-4
up01-5
up01-6
mz02-1
mz02-2
mz02-3
mz02-4
mz02-5
mz02-6
up02-1
up02-2
up02-3
up02-4
up02-5
up02-6
up02-7
mz03-1
mz03-2
mz03-3
mz03-4
mz03-5
up03-1
up03-2
up03-3
up03-4
up03-5
mz04-1
mz04-2
mz04-3
mz04-4
mz04-5
mz04-6
up04-1
up04-2
up04-3
up04-4
up04-5
mz05-1
mz05-2
mz05-3
mz05-4
mz05-5
up05-1
up05-2
up05-3
up05-4
up05-5
mz06-1
mz06-2
mz06-3
mz06-4
mz06-5
up06-1
up06-2
up06-3
up06-4

Submit a solution for up04-5-mz04-5 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M

Problem up04-5: mz04-5 (дореш)

В аргументах командной строки задаются имя входного бинарного файла (`FILE1`), имя выходного бинарного файла (`FILE2`) и положительное 32-битное знаковое целое число (`MOD`). Файл `FILE1` содержит множество положительных целых чисел, представленное в виде битового массива. Бит 0 (младший бит первого байта) в файле соответствует числу 1, бит 7 (старший бит первого байта) в файле соответствует числу 8 и т. д. Установленный бит означает, что соответствующее число присутствует во множестве. Для каждого числа x, присутствующего во множестве, хранящемся в `FILE1`, в бинарный файл `FILE2` запишите 32-битное знаковое целое число $\sum_{i=1}^x i^2 \pmod{MOD}$, то есть остаток от деления на `MOD` суммы квадратов всех чисел от 1 до x включительно. Числа должны быть записаны в порядке возрастания значения x. Числа записываются в порядке байт хоста. Для работы с файлами использовать интерфейс системных вызовов.

Например, если входной бинарный файл содержит байт 0x93, и значение MOD равно 1000, то в выходной бинарный файл должны быть записаны числа 1, 5, 55, 204.

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File Выберите файл Файл не выбран

Send! Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
1562	2022/10/08 10:04:20	2022	up04-5	gcc-32	Pending review	7	**100**	View	View

mz01-1
mz01-2
mz01-3
mz01-4
mz01-5
mz01-6
up01-1
up01-2
up01-3
up01-4
up01-5
up01-6
mz02-1
mz02-2
mz02-3
mz02-4
mz02-5
mz02-6
up02-1
up02-2
up02-3
up02-4
up02-5
up02-6
up02-7
mz03-1
mz03-2
mz03-3
mz03-4
mz03-5
up03-1
up03-2
up03-3
up03-4
up03-5
mz04-1
mz04-2
mz04-3
mz04-4
mz04-5
mz04-6
up04-1
up04-2
up04-3
up04-4
up04-5
mz05-1
mz05-2
mz05-3
mz05-4
mz05-5
up05-1
up05-2
up05-3
up05-4
up05-5
mz06-1
mz06-2
mz06-3
mz06-4
mz06-5
up06-1
up06-2
up06-3
up06-4