



Submit a solution for up16-1-mz16-1 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-12-12 20:00:00
Deadline:	2022-12-16 12:00:00

Problem up16-1: mz16-1 (дореш)

В параметрах командной строки задаются параметры `memory_size`, `cache_size`, `block_size`, определяющие размер ОЗУ, размер кеш-памяти и размер блока кеш-памяти. Размер блока - один из 16, 32, 64 байта. Размер памяти и размер кеша кратны размеру блока.

В кеше прямого отображения (direct mapping) блок ОЗУ может размещаться в единственном блоке кеша, номер которого определяется по формуле `cache_block_num = memory_block_num % block_count`, где `block_count` - число блоков кеша.

На стандартный поток ввода подается трасса выполнения программы. Трасса задается в формате:

R (D|I) ADDR SIZE VALUE

Где R означает чтение, W – запись, D – работу с данными, I – работу с инструкциями, ADDR – 32-битный адрес, задаваемый в 16-ричном виде, SIZE – размер операции (1, 2, 4 байта), VALUE – считываемое значение – число в десятичной системе счисления со знаком соответствующего размера. Например.

RD 00000000 1 0
RD 00000001 1 0
RD 00000002 1 0

На стандартный поток вывода напечатайте число промахов в кеше, причиной которых служит стратегия прямого отображения, то есть число случаев, когда при чтении из памяти блок кеша, в который должен быть загружен блок памяти, уже занят другим блоком памяти.

Submit a solution

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

File Файл не выбран

Send!

View
[all](#) / [necessary](#)

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

up15-1

up15-2

up15-3

up15-4

up15-5

mz16-1

up16-1

up16-2

up16-5



Submit a solution for up16-2-mz16-2 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-12-12 20:00:00
Deadline:	2022-12-16 12:00:00

Problem up16-2: mz16-2 (дореш)

Напишите функцию `calc_max_size` с прототипом:

```
int64_t calc_max_size(int64_t block_size, int64_t block_num_size, int64_t inode_direct_block_count);
```

Параметр `block_size` — это размер блока диска в байтах (положительное число). Параметр `block_num_size` — размер числа для хранения номера блока диска в байтах (положительное число, не больше чем размер блока диска). Параметр `inode_direct_block_count` — число номеров блоков, хранящихся непосредственно в индексном дескрипторе.

Предполагая принцип хранения занятых файлом блоков диска, реализованный в файловых системах Unix (то есть с ссылочными блоками 1-, 2- и 3- уровня косвенности), функция должна вычислить теоретический максимальный поддерживаемый размер файла в байтах. Если размер не представим 64-битным знаковым целым числом, функция должна возвращать значение -1. Накладными расходами на организацию самой файловой системы пренебрегайте.

Напишите программу: программе на стандартный поток ввода задаются три числа: `block_size`, `block_num_size`, `inode_direct_block_count`. Программа на стандартный поток вывода выводит теоретический максимальный поддерживаемый размер файла для заданных входных данных. Затем программа находит такое значение `block_num_size`, при котором теоретический максимальный поддерживаемый размер файла был бы максимальным и выводит найденное значение и соответствующий теоретический максимальный поддерживаемый размер файла.

Максимальный возможный теоретический максимальный поддерживаемый размер файла в терминах возвращаемого функцией `calc_max_size` значения равен $2^{63}-1$, то есть поддерживается файл размера, представимого 64-битным знаковым числом. Если при нескольких размерах `block_num_size` достигается максимум, выведите минимальное.

Examples

Input

```
512 3 10
```

Output

```
2530344960
3 2530344960
```

Submit a solution

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

File Файл не выбран

Send!

View
all / necessary
up02-4
up02-5
up02-6
up02-7
mz09-1
mz09-2
mz09-3
mz09-4
mz09-5
mz09-6
up15-1
up15-2
up15-3
up15-4
up15-5
mz16-2
up16-1
up16-2
up16-5



Submit a solution for up16-3-mz16-3 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-12-12 20:00:00
Deadline:	2022-12-16 12:00:00

Problem up16-3: mz16-3 (дореш)

На стандартном потоке ввода задается состояние таблицы FAT файловой системы.

Первое число N задает число блоков файловой системы ($8 \leq N \leq 2^{20}$). Затем следуют N - 4 записей о состоянии соответствующих блоков Первые два блока не используются для размещения файлов, то есть первый блок, в котором может размещаться файл, имеет номер 2. Последние два блока также не используются для размещения файлов.

Каждая запись о состоянии блока может быть равна 0, что означает, что этот блок свободен, может содержать номер следующего блока файловой системы, может быть равна N - 2, что означает, что данный блок нельзя использовать, или может быть равна N - 1, что обозначает последний блок файла.

Затем на стандартном потоке ввода задаются номера первого блока файла. Для каждого номера первого блока файла на стандартный поток вывода напечатайте размер файла в блоках.

Examples

Input

```
16
15 0 0 8 5 0 2 11 14 15 0 14
6
9
```

Output

```
4
2
```

Submit a solution

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

File Файл не выбран

Send!

View
[all](#) / [necessary](#)

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

up15-1

up15-2

up15-3

up15-4

up15-5

mz16-3

up16-1

up16-2

up16-3

up16-5



09:16:36 / RUNNING

Submit a solution for up16-4-mz16-4 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-12-12 20:00:00
Deadline:	2022-12-16 12:00:00

Problem up16-4: mz16-4 (дореш)

Напишите программу, вычисляющую размер и выравнивание структуры.

На стандартном потоке ввода задается последовательность размеров полей структуры. Требуемое выравнивание для поля равно максимальной степени двойки, на которую без остатка делится размер поля. Например, если размер равен 5, то выравнивание - 1, а если размер равен 6, то выравнивание равно 2.

На стандартный поток вывода напечатайте два числа: размер всей структуры и выравнивание для размещения структуры в памяти. Пустая структура имеет размер 1 и выравнивание 1.

Гарантируется, что размер структуры не превышает UINT32_MAX.

Examples

Input

1 4

Output

8 4

Submit a solution

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

File Файл не выбран

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
5471	2022-12-12 20:29:11	695	up16-4	gcc-32	Pending review	6	100	View	View

View [all](#) / [necessary](#).

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

up15-1

up15-2

up15-3

up15-4

up15-5

mz16-4

up16-1

up16-2

up16-4

up16-5



09:15:29 / RUNNING

Submit a solution for up16-5-mz16-5 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	2 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-12-12 20:00:00
Deadline:	2022-12-16 12:00:00

Problem up16-5: mz16-5 (дореш)

Программе в аргументах командной строки задаются два вещественных (double) значения left и right целое число n и непустое арифметическое выражение func, удовлетворяющее правилам языка Си, которое может содержать одно или несколько вхождений вещественной (double) переменной x. В выражении могут использоваться функции, определенные в заголовочном файле math.h.

Методом прямоугольников вычислите площадь под графиком функции func(x) на отрезке [left;right]. В методе прямоугольников отрезок [left;right] разбивается на n частей, берется значение функции func(x) в левой точке каждого подотрезка, вычисляется площадь прямоугольника, затем эти площади суммируются.

То есть, $dx = (right - left) / n$, $x[i] = left + i * dx$, $s[i] = func(x[i]) * dx$, $s = s[0] + s[1] + ... + s[n - 1]$.

Результат выводите с помощью форматного преобразования %.10g.

Функция func непрерывна на отрезке.

Не используйте system или popen или /bin/sh или аналогичные командные процессоры. Вычисление площади должно выполняться в основном процессе, без вызова других программ с помощью fork или exec, но при подготовке к вычислениям запускать другие программы разрешается.

Учтите, что на тестирующем компьютере работает 64-битная операционная система.

Например, если программа запускается с аргументами

```
./solution 0 1 65536 "sin(x)"
```

Результат ее работы равен:

```
0.4596912742
```

Submit a solution

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

File Файл не выбран

Send!

View all / necessary
up02-4
up02-5
up02-6
up02-7
mz09-1
mz09-2
mz09-3
mz09-4
mz09-5
mz09-6
up15-1
up15-2
up15-3
up15-4
up15-5
mz16-5
up16-1
up16-2
up16-5