Submit a solution for mz03-1-mz03-1

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/19 10:30:00
Deadline:	2022/09/19 12:05:00

Problem mz03-1: mz03-1

Напишите функцию `bit_reverse` со следующим прототипом на Си:

```
STYPE bit_reverse(STYPE value);
```

Где `STYPE` — это некоторый целый знаковый тип. Кроме того, определен тип `UTYPE` — это некоторый целый беззнаковый тип того же размера, что и `STYPE`.

Функция меняет порядок бит в числе на противоположный. Например, рассмотрим 4-битное целое число 0101. Поменяв порядок бит на противоположный, получим число 1010. Однако, если то же число рассматривать как 8-битное, обратный порядок бит будет равен 10100000.

Отрицательные числа представляются в дополнительном коде.

В решении запрещается использовать константы и константные выражения, дающие в качестве результата битность типа, например, `32`, `sizeof (value) * CHAR_BIT` и аналогичные.

Для того, чтобы определить типы `STYPE` и `UTYPE` в своей программе для тестирования, можно использовать `typedef` следующим образом:

```
typedef int STYPE;
typedef unsigned int UTYPE;
```

Сдаваемый на проверку текст должен содержать только функцию `bit_reverse`. Определения типов `STYPE` и `UTYPE` и функция `main` в нем находиться не должны.

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
879	2022/09/26 04:06:00	241	mz03-1	gcc-32	Pending review	4	**100**	View	View
658	2022/09/19 11:24:53	480	mz03-1	gcc-32	Rejected	N/A	N/A	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
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

Problem mz03-2-mz03-2

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/19 10:30:00
Deadline:	2022/09/19 12:05:00

Problem mz03-2: mz03-2

В аргументах командной строки задаются 32-битные знаковые целые десятичные числа.

На стандартный поток вывода напечатайте два числа: сначала сумму положительных аргументов командной строки, затем сумму отрицательных аргументов командной строки. Если чисел соответствующей категории нет, выводите 0.

При обработке аргументов командной строки должны проверяться все типы возможных ошибок, как, например, описано здесь.

Пример запуска программы:

```
./solution 1 -3 5 -7 +12
```

Результат выполнения программы:

```
18
-10
```

Не забывайте выводить символ перехода на новую строку в конце вывода!

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

Problem mz03-3-mz03-3

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/19 10:30:00
Deadline:	2022/09/19 12:05:00

Problem mz03-3: mz03-3

В аргументах командной строки задаются вещественные числа:

- Начальный курс некоторой валюты.
- Изменения к курсу в процентах за один день. (ноль или более раз).

Курс валюты задается с четырьмя дробными цифрами, например, 69.5634. Изменения к курсу задаются в процентах и могут быть как положительными, так и отрицательными. Например, изменение курса -10.0 означает, что за день валюта подешевела на 10%, то есть, если в начале дня курс был 100.0000, то в конце дня он окажется 90.0000. Курс валюты в конце дня фиксируется с четырмя цифрами дробной части округлением по математическим правилам, все цифры дробной части после четвертой после округления отбрасываются.

Гарантируется, что ежедневные изменения курса больше -100 и меньше 100. Изменения курса задаются не более чем с четырьмя знаками дробной части. Курс валюты — положительный и не превосходит 10000 за все время.

На стандартный поток вывода напечатайте курс валюты в конце последнего дня, заданного в командной строке. Курс печатайте с четырьмя знаками дробной части.

Пример запуска программы:

```
./solution 100.0 -10.0 -5.5 1.0
```

Результат выполнения программы:

```
85.9005
```

Не забывайте выводить символ перехода на новую строку в конце вывода!

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

Problem mz03-4-mz03-4

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/19 10:30:00
Deadline:	2022/09/19 12:05:00

Problem mz03-4: mz03-4

Напишите функцию

```
int satsum(int v1, int v2);
```

которая выполняет сложение с насыщением двух знаковых чисел. То есть, если результат сложения не представим типом результата, результатом в зависимости от знака будет либо минимальное, либо максимальное число, представимое типом результата. Например, если складываются с насыщением два четырехбитных числа (диапазон представимых чисел [−8;7]): 5 + 5 => 7, −4 + −6 => −8.

Тип `int` имеет размер 32 бита. Отрицательные числа представляются в дополнительном коде.

В функции должны быть объявлены две константы:

```
enum { MY_INT_MAX = ... };  // максимальное значение знакового 32-битного типа
enum { MY_INT_MIN = ... };  // минимальное значение знакового 32-битного типа
```

Разрешается использовать только 32-битные целые типы. Разрешается использовать только константу 0. Разрешается использовать только стандартные средства языка Си (расширения gcc использовать запрещено). Константы из стандартной библиотеки (`INT_MAX` и аналогичные) также использовать запрещено.

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
692	2022/09/19 12:02:53	422	mz03-4	gcc-32	Ignored	N/A	N/A	View	N/A
691	2022/09/19 12:02:21	510	mz03-4	gcc-32	Compilation error	N/A	N/A	View	View
682	2022/09/19 11:52:38	421	mz03-4	gcc-32	Partial solution	0	**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
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

Problem mz03-5-mz03-5

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/09/19 10:30:00
Deadline:	2022/09/19 12:05:00

Problem mz03-5: mz03-5

Элемент списка определяется следующим образом:

```
struct Elem
{
    struct Elem *next;
    char *str;
};
```

`str` указывает на строку, размещенную в области динамической памяти.

Напишите функцию на Си:

```
struct Elem *dup_elem(struct Elem *head);
```

Если у некоторого элемента `p` в строке `p->str` записано текстовое десятичное представление 32-битного целого числа x, причем пробелы в начале допускаются, а в конце - нет, то перед каждым таким элементом в список добавляется новый элемент, у которого в строке записано десятичное представление числа x + 1, если оно представимо 32-битным целым знаковым значением. В текстовом представлении числа в новом элементе списка не должно быть пробелов, лишних знаков плюс и нулей.

Функция должна вернуть указатель на первый элемент получившегося списка.

Например, если в списке хранились строки "10" "5x" "alpha" " -03", то в результирующем списке должны храниться строки "11" "10" "5x" "alpha" "-2" " -03".

В сдаваемой на проверку программе должно присутствовать определение структуры `struct Elem`.

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