



10:02:49 / RUNNING

Submit a solution for up07-1-mz07-1 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/17 20:00:00
Deadline:	2022/11/07 20:00:00
Date penalty formula:	2022/12/31 0-50/7d
Current penalty:	50
Next soft deadline:	2022/10/31 20:00:00

Problem up07-1: mz07-1 (дореш)

Рассмотрим троичную симметричную систему счисления. В троичной симметричной системе счисления используются три цифры '0', '1', 'a'. Цифра 'a' имеет значение -1, веса позиций являются степенью числа 3, как обычно. Число 'a1' равно $-1 \cdot 3 + 1 = -2$ в десятичной системе счисления.

На стандартном потоке ввода задается последовательность чисел, записанных в троичной симметричной системе счисления. Числа разделяются произвольным количеством пробельных символов. Перед первым числом и после последнего числа может следовать произвольное количество пробельных символов.

Если очередное число представимо 64-битным знаковым двоичным числом, на стандартный вывод напечатайте его в десятичном виде. Если очередное число непредставимо 64-битным знаковым двоичным числом, на стандартный вывод напечатайте '18446744073709551616'. Каждое число выводите на отдельной строке.

Не забывайте выводить `\n` в конце вывода.

Examples

Input

```
a a1 10a
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Output

```
-1
-2
8
18446744073709551616
```

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
3214	2022/10/24 19:49:33	1473	up07-1	gcc-32	Pending review	12	100	View	View

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

up02-4

up02-5

up02-6

up02-7

up07-1

mz08-1

mz08-2

mz08-3

mz08-4

mz08-5



10:03:08 / RUNNING

Submit a solution for up07-2-mz07-2 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/17 20:00:00
Deadline:	2022/11/07 20:00:00
Date penalty formula:	2022/12/31 0-50/7d
Current penalty:	50
Next soft deadline:	2022/10/31 20:00:00

Problem up07-2: mz07-2 (дореш)

В аргументах командной строки задается последовательность операндов. Каждый корректный операнд представляет собой знаковое 32-битное число, непосредственно после которого идет знак “+” или “-”, который обозначает операцию, которую нужно выполнить. Само число может содержать суффикс ‘k’, обозначающий, что значение нужно умножить на 1000. Например, последовательность операндов

+100+ -20- 5k+

означает, что вычисляется выражение 0 + (+100) - (-20) + (5000).

Некорректно записанные операнды, или операнды, у которых значение не является представимым знаковым 32-битным числом, должны игнорироваться.

На стандартный поток вывода напечатайте результат вычисления выражения.

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

Пример запуска программы:

./solution +100+ -20- 5k+

Результат работы:

5120

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
2967	2022/10/24 09:02:12	832	up07-2	gcc-32	Pending review	8	50=100-10*10	View	View
2966	2022/10/24 09:01:33	697	up07-2	gcc-32	Ignored	N/A	N/A	View	N/A
2965	2022/10/24 09:00:28	694	up07-2	gcc-32	Partial solution	0	0=0-9*10	View	View
2964	2022/10/24 08:59:47	674	up07-2	gcc-32	Partial solution	0	0=0-8*10	View	View
2960	2022/10/24 04:41:59	787	up07-2	gcc-32	Partial solution	6	0=0-7*10	View	View
2959	2022/10/24 04:40:56	812	up07-2	gcc-32	Partial solution	6	0=0-6*10	View	View
2958	2022/10/24 04:24:51	997	up07-2	gcc-32	Partial solution	4	0=0-5*10	View	View
2957	2022/10/24 04:22:51	997	up07-2	gcc-32	Partial solution	0	0=0-4*10	View	View
2956	2022/10/24 04:15:04	791	up07-2	gcc-32	Partial solution	6	0=0-3*10	View	View
2955	2022/10/24 04:00:51	749	up07-2	gcc-32	Partial solution	6	0=0-2*10	View	View
2954	2022/10/24 03:51:16	731	up07-2	gcc-32	Partial solution	6	0=0-1*10	View	View
2953	2022/10/24 03:49:36	862	up07-2	gcc-32	Partial solution	6	0	View	View

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

up02-4

up02-5

up02-6

up02-7

up07-2

mz08-1

mz08-2

mz08-3

mz08-4

mz08-5



10:03:25 / RUNNING

Submit a solution for up07-3-mz07-3 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/17 20:00:00
Deadline:	2022/11/07 20:00:00
Date penalty formula:	2022/12/31 0-50/7d
Current penalty:	50
Next soft deadline:	2022/10/31 20:00:00

Problem up07-3: mz07-3 (дореш)

Напишите программу, которая во втором каталоге найдет клоны файлов из первого каталога. Под клоном файла понимаются другие имена исходного файла или симлинки на исходный файл или другие имена исходного файла.

В аргументах командной строки передаются пути к двум каталогам. На стандартный поток вывода напечатайте суммарный размер тех регулярных файлов из первого каталога, которые могут быть модифицированы текущим пользователем, и клон которого под тем же самым именем находятся во втором каталоге. При этом симлинки в первом каталоге игнорируются, но во втором каталоге файл с тем же самым именем может быть и симлинком на файл в первом каталоге.

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
2952	2022/10/24 03:21:40	1899	up07-3	gcc-32	Pending review	6	100	View	View

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

up02-4

up02-5

up02-6

up02-7

up07-3

mz08-1

mz08-2

mz08-3

mz08-4

mz08-5



Submit a solution for up07-4-mz07-4 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/17 20:00:00
Deadline:	2022/11/07 20:00:00
Date penalty formula:	2022/12/31 0-50/7d
Current penalty:	50
Next soft deadline:	2022/10/31 20:00:00

Problem up07-4: mz07-4 (дореш)

В аргументе командной строки задается путь к существующему регулярному бинарному файлу. Файл непустой и его размер кратен 8. Файл содержит битовую маску (битовое множество, bitset) чисел. Байт содержит 8 бит. Биты хранятся в порядке Little-Endian и нумеруются с 1, то есть биту с номером 1 соответствует младший бит байта со смещением 0 в файле, биту с номером 16 соответствует старший бит байта со смещением 1 в файле, и т.д.

На стандартном потоке ввода подается последовательность 32-битных знаковых целых чисел. Если очередное число X положительное и бит с номером X не выходит за пределы файла, то бит X в файле устанавливается. Если число X отрицательное и бит с номером -X не выходит за пределы файла, бит X в файле сбрасывается. Все прочие значения игнорируются.

Можете предполагать, что хост процессор имеет порядок байт Little-Endian.

Хранить в памяти более 8 байт содержимого файла, либо использовать mmap запрещено.

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
2971	2022/10/24 09:21:34	1811	up07-4	gcc-32	Pending review	3	80=100-2*10	View	View
2970	2022/10/24 09:11:22	1707	up07-4	gcc-32	Partial solution	1	0=0-1*10	View	View
2961	2022/10/24 05:38:27	1474	up07-4	gcc-32	Partial solution	0	0	View	View

View all / necessary
up02-4
up02-5
up02-6
up02-7
up07-4
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5



10:04:04 / RUNNING

Submit a solution for up07-5-mz07-5 (дореш)

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/17 20:00:00
Deadline:	2022/11/07 20:00:00
Date penalty formula:	2022/12/31 0-50/7d
Current penalty:	50
Next soft deadline:	2022/10/31 20:00:00

Problem up07-5: mz07-5 (дореш)

Некоторое событие традиционно проводится один раз в год каждый год. Дата проведения определяется как четвертый понедельник строго после первого дня полнолуния строго после 256 дня в году. Днем полнолуния считаются сутки по UTC в которых находится момент полнолуния. Все даты рассчитываются по UTC. Например, один из моментов полнолуния в 2021 году произошел 26 мая 2021 года в 11:14:00 UTC. Синодический период обращения Луны (то есть интервалы между моментами полнолуния) равен 29 дней 12 часов 44 минуты.

На стандартном потоке ввода подается одно целое число — год в интервале от 1902 до 2037 года включительно. На стандартный поток вывода напечатайте дату события в этот год в формате YYYY-MM-DD.

Examples

Input

2021

Output

2021-10-18

Input

2022

Output

2022-11-07

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
2945	2022/10/24 02:11:03	1828	up07-5	gcc-32	Pending review	11	100	View	View

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

up02-4

up02-5

up02-6

up02-7

up07-5

mz08-1

mz08-2

mz08-3

mz08-4

mz08-5