



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings

Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:28:40 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ1-Примеры на сложение и умножение

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Примеры на сложение и умножения

Представьте, что вы учитель математики в школе и вам надо проверить правильно ли ученики решают примеры на сложение умножение.

Итак, необходимо написать программу, которой на стандартный ввод поступает пример на сложение или умножение и проверить верн или неверно пример решен.

Однако надо помнить, что этой программой будут пользоваться ученики в школе, а они могут случайно или специально написать прим неверно (забыли написать знак '=' или забыли написать ответ или все перепутали и написали вместо '=' другой знак) Программа должн корректно обрабатывать все ошибочные ситуации.

При этом в примере числа и знаки '+', '*' или '=' могут быть окружены произвольным числом пробелов или символов табуляции и это и считается ошибкой.

Input format

%d*%d=%d или %d+%d=%d Но возможен ввод произвольных текстовых символов вместе с произвольным числом символов табуляции.

Output format

- Если пример решен верно вывести: YES
- Если пример решен неверно вывести: NO
- Если пример не имеет числа после знака '=', вывести: NO ANSWER
- Если пример после первого числа не содержит знака '+' или '*' вывести: WRONG SIGN
- Если пример после второго числа не содержит знака '=' вывести: NO EQUAL SIGN
- Во всех остальных случаях вывести: INPUT ERROR

Examples

Input

23*2=46

Output

YES

Input

456+123=789

Output

NO

Input

67-34=23

Output

WRONG SIGN

Input

$56 * 3 = 150$

Output

NO

Notes

В случае нескольких ошибок в одном примере выводить первую ошибку. Например, если пример не содержит '*' или '+' и не содержит знака '=', то вывод WRONG SIGN потому что в примере знак '+' или '*' находится перед знаком '='.

Следующая таблица поможет понять правила вывода: (все примеры со знаком '+' для знака '*' все аналогично)

- $1+1=2$ - YES
- $1+1=0$ - NO
- $a1+1=2$ - INPUT ERROR
- $1a+1=2$ - WRONG SIGN
- $1+a1=2$ - INPUT ERROR
- $1+1a=2$ - NO EQUAL SIGN
- $1+1=a2$ - NO ANSWER
- $1+1=2a$ - INPUT ERROR
- 1 - WRONG SIGN
- $1+$ - INPUT ERROR
- $1+1=$ - NO ANSWER

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosenSend!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
34	2022/09/24 19:13:59	1972	DZ1	gcc	OK	N/A	View	View
33	2022/09/24 19:10:26	1930	DZ1	gcc	Wrong answer	35	View	View
32	2022/09/24 19:04:52	1921	DZ1	gcc	Wrong answer	33	View	View
31	2022/09/24 19:00:58	1919	DZ1	gcc	Wrong answer	5	View	View

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings

Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:29:17 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ2-Битовая упаковка

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Битовая упаковка

Написать программу-упаковщик чисел.

Пользователь вводит количество чисел для упаковки (не более 255) и сами числа (от 0 до 255)

Упаковать необходимо в массив типа char следующего формата:

- нулевой элемент - количество чисел,
- следующие элементы (обозначим их количество за k) предназначены для хранения количества значащих битов в упаковываемых числах: (старшие полбайта первого элемента массива относятся к 1му числу, младшие полбайта - ко 2му; старшие полбайта второго элемента массива относятся к 3му числу и т.д.; если количество упаковываемых чисел нечётное - соответствующие полбайта содержат нули)
- оставшиеся элементы массива содержат упакованные значащие биты, начало этой битовой последовательности должно быть прижато левому краю k+2-го элемента массива

Программа-упаковщик должна быть оформлена в виде функции, которая принимает в качестве параметров указатель на массив введённых чисел (тип char*), количество элементов, другие необходимые параметры. Возвращать функция должна указатель на массив упакованных чисел (тип char *). Память под массив исходных чисел выделяется в main(), память под массив упаковки - внутри функции. Эти массивы должны быть динамическими. Функцию realloc() использовать запрещено.

Пример. Пользователь вводит числа 2, 3 и 253. Таким образом, первое число 2 - это количество чисел, а 3 253 сами числа. Бинарные записи этих чисел: 2 - 10, 3 - 11, 253 - 1111 1101. Количества значащих бит в числах 3 и 253: 2(10) и 8 (1000). Массив упаковки будет выглядеть следующим образом: 0000 0010 | 0010 1000 | 1111 1111 | 0100 0000

Input format

Последовательность целых чисел

Output format

последовательность бит упакованного массива в виде 0 и 1, отделяемых пробелом через каждые 8 элементов

Examples

Input

2 3 253

Output

00000010 00101000 11111111 01000000

Notes

У числа 0 число кодирующих значащих бит считаем равным 0, а в упаковочные значащие биты вообще ничего не добавляем. Пример: 4 0 1 2 3. Будет закодировано 0000 0100 | 00 0001 | 0010 0010 | 1101 1000.

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosenSend!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
127	2022/10/02 00:59:04	4009	DZ2	gcc	OK	N/A	View	View
126	2022/10/02 00:54:22	3996	DZ2	gcc	Wrong answer	1	View	View
125	2022/10/02 00:35:53	3854	DZ2	gcc	Wrong answer	4	View	View

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings

Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:29:52 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ3-Работа со строками

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Работа со строками

1. Реализовать собственные аналоги функций `strlen`, `strcmp`, `strstr`. Назвать их `my_strlen`, `my_strcmp`, `my_strstr`. Интерфейс функций (аргумент и возвращаемое значение) должен быть такой же, как у аналогичных функций из `string.h`. Использовать сами функции из `string` запрещено.
2. Реализовать функции `print_str_arr` (распечатка массива строк `char **`) и `search_arr` (поиск заданного количества вхождений некоторой подстроки в массиве строк `char **`; использовать свои функции из п.1; использовать функции из `string.h` запрещено). Функция `search_arr` должна распечатать все строки из массива строк `char **` (вместе с их номерами в массиве), в которых заданная подстрока встречается заданное число раз.
3. Написать программу, которая
 - читает из стандартного ввода несколько строк,
 - сохраняет эти строки в массив `char **`,
 - читает из командной строки подстроку и необходимое количество вхождений,
 - с помощью `search_arr` распечатывает все строки из созданного массива строк (вместе с их номерами в массиве), в которых заданная подстрока встречается заданное число раз.
4. Функции из п.1,2 должны корректно обрабатывать нештатные ситуации (на вход подали указатель `NULL`, указали отрицательное или нулевое кол-во вхождений подстроки и т.д.) Функция `my_strlen` от `NULL` должна возвращать минус 1. Функция `my_strstr` возвращает `NULL` если один из аргументов `NULL`. Функция `my_strcmp` возвращает `INT_MAX`, если один из аргументов `NULL`.
5. Число строк и количество символов в каждой строке заранее неизвестно и может быть любым.

Input format

Текст подается на стандартный ввод. В командной строке два параметра: подстрока и число вхождений

Output format

%d:%s где %d номер строки в массиве, (нумерация с 1) а %s сама строка

Examples

Input

```
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles
And by opposing end them, to die – to sleep,
No more; and by a sleep to say we end
The heart-ache and the thousand natural shocks
That flesh is heir to: 'tis a consummation
Devoutly to be wish'd. To die, to sleep;
```

Command line: to 2

Output

```
5:And by opposing end them, to die – to sleep,
9:Devoutly to be wish'd. To die, to sleep;
```

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosen

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
308	2022/10/26 12:41:42	3401	DZ3	gcc	OK	N/A	View	View
266	2022/10/19 15:19:27	3401	DZ3	gcc	Ignored	N/A	View	N/A
265	2022/10/19 15:12:08	3164	DZ3	gcc	Ignored	N/A	View	N/A
261	2022/10/19 11:38:26	3136	DZ3	gcc	Wrong answer	4	View	View
246	2022/10/18 20:18:13	3129	DZ3	gcc	Wrong answer	4	View	View
244	2022/10/18 20:01:31	3145	DZ3	gcc	Wrong answer	4	View	View
241	2022/10/17 17:49:18	3145	DZ3	gcc	Wrong answer	4	View	View
235	2022/10/13 20:04:06	2732	DZ3	gcc	Wrong answer	4	View	View
234	2022/10/13 19:30:05	2769	DZ3	gcc	Run-time error	3	View	View
232	2022/10/13 19:22:41	2699	DZ3	gcc	Run-time error	3	View	View
208	2022/10/12 12:13:25	2700	DZ3	gcc	Run-time error	3	View	View
204	2022/10/11 21:27:16	2702	DZ3	gcc	Run-time error	1	View	View
203	2022/10/11 21:15:49	2675	DZ3	gcc	Run-time error	1	View	View

[DZ1](#)[DZ2](#)[DZ3](#)[DZ5](#)[DZ6](#)[DZ7](#)

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings
Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:30:20 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ5-Обработка файла

Time limit: 2 s

Real time limit: 5 s

Memory limit: 64M

Обработка текста

Написать программу обработки текста. Предполагается, что текст содержится в файле, располагающемся в той же папке, что программа. Программа дублирует часть букв каждого слова в соответствии с заданными аргументами командной строки.

Разделителями считаются пробел, точка и запятая.

Первый аргумент командной строки - имя файла с текстом. Далее могут идти:

- `rep tail n` - дублирует (т.е. приписывает в конце) в каждом слове `n` последних букв. Если длина слова меньше `n` - оно остаётся без изменений. Пример: при запросе `rep tail 2` слово `dog` превратится в `dogog`
- `rep head n` - дублирует (т.е. приписывает в начало) в каждом слове `n` первых букв. Если длина слова меньше `n` - оно остаётся без изменений. Пример: при запросе `rep head 3` слово `horse` превратится в `horhorse`
- `rep ends n m` - дублирует в каждом слове `n` первых и `m` последних букв. Если длина слова меньше `n+m` - оно остаётся без изменений. Пример: при запросе `rep ends 2 3` слово `сгосо` превратится в `сгсгосоосо`

Запрещается использовать функции из `string.h` и считывать файл целиком. Запрещается использовать дополнительные файлы, результат сохраняется в исходный файл. Замечание: для работы с файлами использовать только низкоуровневый ввод-вывод.

Examples

Input in data.txt

```
Hello world,  
my dear world.
```

Output in data.txt

```
HeHelloo woworldd,  
my dedearr woworldd.
```

Notes

Пример вызова программы: `a.out data.txt rep ends 2 1`

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosen

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
342	2022/11/14 23:36:57	3177	DZ5	gcc	OK	N/A	View	View
341	2022/11/14 17:44:41	3161	DZ5	gcc	Wrong answer	8	View	View
340	2022/11/14 17:37:11	3034	DZ5	gcc	Time-limit exceeded	1	View	View
339	2022/11/14 00:59:44	2988	DZ5	gcc	Wrong answer	1	View	View
338	2022/11/14 00:47:00	2964	DZ5	gcc	Wrong answer	8	View	View
337	2022/11/14 00:46:25	3	DZ5	gcc	Compilation error	N/A	View	View
336	2022/11/14 00:42:34	3000	DZ5	gcc	Wrong answer	8	View	View
335	2022/11/14 00:36:29	2926	DZ5	gcc	Wrong answer	8	View	View

334	2022/11/13 23:31:52	2828	DZ5	gcc	Wrong answer	1	View	View
-----	---------------------	------	-----	-----	--------------	---	----------------------	----------------------

DZ1	DZ2	DZ3	DZ5	DZ6	DZ7
-----	-----	-----	-----	-----	-----

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings

Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:30:39 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ6-Конвейер из N процессов

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Конвейер из N процессов

Реализовать простейший вариант командной оболочки shell.

Программа **на стандартный поток ввода** получает конвейер из N ступеней, т.е. строку вида:

команда_1 [параметры] | ... | команда_N [параметры]

и выполняет её. Здесь $N \geq 1$. При $N = 1$ конвейер вырождается в одну команду с возможными параметрами. Строка всегда заканчивается символом перевода строки.

Параметром является либо последовательность текстовых непробельных символов, либо последовательность текстовых (в том числе пробельных!) символов заключенных в одинарные или двойные кавычки, в этом случае все что находится внутри кавычек считается одним параметром. Под текстовым символом понимается любой печатный символ, кроме символов кавычек и символа '|'

Предусмотреть возможные ошибочные ситуации:

- вызов несуществующей команды, в этом случае вывод: "Command %s not found", где %s имя несуществующей команды. Если в команде две и более несуществующих команд, то осуществлять вывод первой несуществующей команды, при этом конвейер прерывается оставшаяся часть не выполняется.
- ошибка в синтаксисе команды: две вертикальные палочки подряд, например `ls || wc`, или отсутствие команды после вертикальной палочки, например, `ls|`, или нечетным количеством кавычек и т.д. Во всех таких случаях вывод "Syntax error"

Все выводы ошибочных ситуаций выводятся в **поток ошибок**.

Input format

команда_1 [параметры] | ... | команда_N [параметры]

Output format

Вывод, соответствующий команде конвейера shell в stdout, либо соответствующей ошибки в stderr

Examples

Input

```
ls -l -a | sort -k 2 | grep -v rwx | wc
```

Output

```
14      110      810
```

Notes

Обратите внимание, числа в выводе примера зависят от количества и свойств файлов в вашей папке, поэтому вероятней всего у вас будут другие значения.

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosen

Send!

Previous submissions of this problem (last 15)

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
--------	------	------	---------	----------	--------	-------------	-------------	-------------

Run ID	Time	Size	Problem	Language	Result	Failed test	view source	view report
619	2022/12/20 16:16:46	17824	DZ6	gcc	OK	N/A	View	View
587	2022/12/19 20:59:33	16473	DZ6	gcc	Ignored	N/A	View	N/A
581	2022/12/19 19:43:23	15350	DZ6	gcc	Wrong answer	17	View	View
571	2022/12/18 22:20:51	15100	DZ6	gcc	Wrong answer	17	View	View
530	2022/12/18 14:12:54	14405	DZ6	gcc	Ignored	N/A	View	N/A
517	2022/12/18 02:12:32	14034	DZ6	gcc	Wrong answer	18	View	View
407	2022/11/23 19:13:23	5938	DZ6	gcc	OK	N/A	View	View
406	2022/11/23 18:40:23	5278	DZ6	gcc	Wrong answer	17	View	View
401	2022/11/22 20:17:33	11615	DZ6	gcc	Wrong answer	15	View	View
398	2022/11/21 23:34:12	10700	DZ6	gcc	Wrong answer	15	View	View
397	2022/11/21 23:27:34	10665	DZ6	gcc	Wrong answer	15	View	View
396	2022/11/21 23:23:18	10780	DZ6	gcc	Wrong answer	15	View	View
395	2022/11/21 23:16:27	10742	DZ6	gcc	Wrong answer	15	View	View
394	2022/11/21 23:00:03	10305	DZ6	gcc	Wrong answer	15	View	View
393	2022/11/21 18:55:37	9811	DZ6	gcc	Wrong answer	15	View	View

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217]: Submit a solution

Settings

Logout [cmc_217_2022_07]

Info

Summary

Submissions

Standings

Submit clar

Clars

21:30:56 / RUNNING

DZ1

DZ2

DZ3

DZ5

DZ6

DZ7

Submit a solution for DZ7-Командный интерпретатор

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Командный интерпретатор (Shell)

В этом задании необходимо расширить функционал, реализованный в предыдущем задании.

1. Программа теперь должна уметь выполнять несколько командных строк, одну за другой, пока пользователь не даст команду `exit`, которая означает выход из shell. Считывание командных строк (конвейеров) происходит в цикле. Каждая строка заканчивается символом `'\n'`. После того, как отработало выполнение одной командной строки, программа ожидает от пользователя ввода следующей.

Завершение shell происходит либо по закрытию стандартного потока ввода, либо ввода пользователем команды `exit`. В первом случае шелл ничего не выводит перед закрытием, во втором - пишет "Shell is closing, bye" и закрывается,

2. Необходимо добавить возможность перенаправления потоков. Теперь общий вид строки команд должен быть следующим

команда_1 [параметры] [перенаправление] | ... | команда_n [параметры] [перенаправление]

Должно корректно обрабатываться перенаправление:

- потока ввода (<)
- потока вывода(>)
- потока вывода с добавлением в конец файла(>>)
- потока ошибок(2>)

Параметром является либо последовательность текстовых непробельных символов, либо последовательность текстовых (в том числе пробельных!) символов заключенных в одинарные или двойные кавычки, в этом случае все что находится внутри кавычек считается одним параметром. Под текстовым символом понимается любой печатный символ, кроме символов кавычек, символа `'|'` и символов `'<'`, `'>'`. На каждой ступени конвейера может одновременно использоваться перенаправление нескольких различных потоков, например, поток ввода и вывода, но использование на одной ступени конвейера перенаправление несколько раз одного и того же потока считается синтаксической ошибкой.

3. Реализовать собственные варианты следующих встроенных команд:

- `cd` - смена текущего каталога
- `pwd` - выдача текущего каталога в стандартный поток вывода
- `exit` - выход из shell. Перед выходом выводится "Shell is closing, bye"

В случае если `exit` появляется на одной из ступеней конвейера, выполнение конвейера прерывается, выводится сообщение перед выходом и shell завершается.

4. Предусмотреть возможные ошибочные ситуации:

- вызов несуществующей команды, в этом случае вывод: "Command %s not found", где %s имя несуществующей команды
- ошибка в синтаксисе команды: неправильное использование символов конвейера и перенаправлений, например `ls >| wc`, использованы после символа перенаправления более одного имени файла, нечетное количество кавычек и т.д. Во всех таких случаях вывод "Syntax error"
- ошибка при открытии файла, в этом случае вывод "File %s not found", где %s имя файла, который не удастся открыть.
- ошибка при открытии директории, в этом случае вывод: "Directory %s not found", где %s имя директории, в которую не удастся войти или которую не удастся прочитать

Все выводы ошибочных ситуаций выводятся в **поток ошибок**.

Input format

команда_1 [параметры] [перенаправление] | ... | команда_n [параметры] [перенаправление]

команда_1 [параметры] [перенаправление] | ... | команда_n [параметры] [перенаправление]

...

команда_1 [параметры] [перенаправление] | ... | команда_n [параметры] [перенаправление]

Output format

Вывод, соответствующих команд shell в stdout, либо соответствующей ошибки в stderr

Examples

Input

```
ls -l -a > ls.txt
sort -k 2 <ls.txt | grep -v rwx | wc
exit
```

Output

```
      14      110      810
Shell is closing, bye
```

Notes

Обратите внимание, числа в выводе примера зависят от количества и свойств файлов в вашей папке, поэтому вероятней всего у вас будут другие значения.

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosen

Send!

Previous submissions of this problem (last 15)

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
621	2022/12/20 17:00:18	18265	DZ7	gcc	Wrong answer	6	View	View
620	2022/12/20 16:49:56	18130	DZ7	gcc	Wrong answer	2	View	View
618	2022/12/20 16:15:11	17824	DZ7	gcc	Wrong answer	2	View	View
617	2022/12/20 16:00:56	17729	DZ7	gcc	Wrong answer	2	View	View
616	2022/12/20 15:41:06	17655	DZ7	gcc	Wrong answer	2	View	View
614	2022/12/20 13:50:41	17558	DZ7	gcc	Wrong answer	2	View	View
613	2022/12/20 13:43:24	17554	DZ7	gcc	Wrong answer	2	View	View
612	2022/12/20 13:36:26	17544	DZ7	gcc	Wrong answer	2	View	View
611	2022/12/20 13:15:56	17542	DZ7	gcc	Wrong answer	2	View	View
607	2022/12/20 01:09:45	17385	DZ7	gcc	Wrong answer	2	View	View
603	2022/12/20 00:48:52	16594	DZ7	gcc	Wrong answer	2	View	View
592	2022/12/19 21:51:45	16365	DZ7	gcc	Wrong answer	2	View	View
590	2022/12/19 21:25:50	15885	DZ7	gcc	Wrong answer	2	View	View
586	2022/12/19 20:58:02	16473	DZ7	gcc	Wrong answer	2	View	View
585	2022/12/19 20:46:49	16429	DZ7	gcc	Wrong answer	1	View	View

DZ1 DZ2 DZ3 DZ5 DZ6 **DZ7**

[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\)](#).

Copyright © 2000-2020 Alexander Chernov.