



21:33:43 / 7 unread message(s) / RUNNING

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

Submit a solution for MZ1-Точка и прямоугольник

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Точка и прямоугольник

На стандартный ввод подается координаты точки (x и y) и две координаты **противоположных** вершин прямоугольника, (x1 y1 x2 y2) стороны которого параллельны осям координат. Порядок выбора вершин не определен. Если точка принадлежит прямоугольнику вывести YES, если нет вывести NO.

Input format

6 вещественных чисел

Output format

YES или NO

Examples

Input

-0.5 1.5 -1.5 1 2 -0.5

Output

NO

Notes

Если точка принадлежит границе прямоугольника, то она принадлежит прямоугольнику

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
7	2022/09/14 16:17:09	477	MZ1	gcc	OK	N/A	View	View

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------



Задворный Даниэль [MSU CMC 217]: Submit a solution

[Settings](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:34:03 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ2-Тигрокрыс

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Тигрокрыс

В одной далекой галактике на планете Пенелопа живет странное животное тигрокрыс, у него два хвоста. Написать программу, которая проверяет правильно ли задан каждый из дгх хвостов и выводит какой хвост длиннее. Хвост задан правильно, если он состоит только из символов "-". Если оба хвоста заданы правильно, то вывести какой длиннее. То есть ес первый хвост длиннее, то вывести "First", если второй хвост длиннее, то вывести "Second". Если хвосты имеют одинаковую длину, то вывести "Equal". Если хотя бы один хвост зад неверно, вывести "Incorrect".

Input format

Две строки, состоящих из произвольных текстовых символов

Output format

First, Second, Equal или Incorrect

Examples

Input

```
=====
```

Output

```
First
```

Input

```
=====
```

Output

```
Incorrect
```

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
95	2022/09/21 13:19:45	864	MZ2	gcc	OK	N/A	View	View
93	2022/09/21 13:18:38	858	MZ2	gcc	Compilation error	N/A	View	View
70	2022/09/21 12:47:18	982	MZ2	gcc	Rejected	N/A	View	View
69	2022/09/21 12:45:25	950	MZ2	gcc	Wrong answer	1	View	View
66	2022/09/21 12:39:02	768	MZ2	gcc	Wrong answer	10	View	View
63	2022/09/21 12:35:00	683	MZ2	gcc	Wrong answer	10	View	View

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-------------	------

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

Copyright © 2000-2020 Alexander Chernov.



21:34:22 / 7 unread message(s) / RUNNING

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

Submit a solution for MZ3-Узлы сетки

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Узлы сетки

Дан квадрат, расположенный в 1-й четверти координатной плоскости. Левый нижний угол лежит в начале координат, нижняя и левые стороны на осях X и Y соответственно. Длина стороны квадрата равна 1.

На нижней стороне квадрата с равными интервалами располагается Nx точек, первая - в 0, последняя - в 1. Аналогично, на левой стороне с равными интервалами располагается Ny точек, первая - в 0, последняя - в 1.

Через точки на нижней стороне проводятся вертикальные прямые до пересечения с верхней стороной. Через точки на левой стороне проводятся горизонтальные прямые до пересечения с правой стороной. Точки пересечения этих прямых назовём узлами.

Необходимо распечатать координаты (x,y) всех узлов. Сначала в строчку печатаются узлы, находящиеся на верхней стороне квадрата. В следующей строке - узлы следующего ряда. И так далее, последними печатаются узлы с нижней стороны квадрата.

Nx и Ny задаются через стандартный ввод.

Input format

Два целых числа Nx и Ny

Output format

Точку выводить в формате (x, y) числа x и y должны содержать ровно 3 знака после точки.

Examples

Input

```
4 5
```

Output

```
(0.000, 1.000) (0.333, 1.000) (0.667, 1.000) (1.000, 1.000)
(0.000, 0.750) (0.333, 0.750) (0.667, 0.750) (1.000, 0.750)
(0.000, 0.500) (0.333, 0.500) (0.667, 0.500) (1.000, 0.500)
(0.000, 0.250) (0.333, 0.250) (0.667, 0.250) (1.000, 0.250)
(0.000, 0.000) (0.333, 0.000) (0.667, 0.000) (1.000, 0.000)
```

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
84	2022/09/21 13:05:18	275	MZ3	gcc	OK	N/A	View	View

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

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

Copyright © 2000-2020 Alexander Chernov.



21:34:36 / 7 unread message(s) / RUNNING

[MZ0](#) [MZ1](#) [MZ2](#) [MZ3](#) [MZ4](#) [MZ5](#) [MZ6](#) [MZ7](#) [MZ8](#) [MZ9](#) [MZ10](#) [MZ11](#) [MZ12](#)

Submit a solution for MZ4-Таблица умножения

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Таблица умножения

На стандартном потоке ввода подаются числа a , b и n . Числа a и b представимы типом `int`. Число n представимо типом `int` и больше 0.

Число n задает ширину ячейки таблицы в символах. Числа a и b – это границы диапазона целых чисел $[a;b]$. Гарантируется, что $a \leq b$.

На стандартный поток вывода напечатайте таблицу умножения для этих чисел как показано в примере. Ячейки таблицы, расположенные на одной строке, разделяются строго одним пробелом. В конце каждой строки выводится символ `\n`. Пробелы в конце строки должны отсутствовать. Гарантируется, что ширина ячейки таблицы достаточна для вывода произведения соответствующих строк и столбца.

Вывод вашей программы будет сравниваться с правильным ответом побайтово. Все пробельные символы – значимы.

Указание: для вывода числа с требуемым количеством пробелов используйте спецификацию * ширины поля вывода `printf`. Например: `printf("%*d", n, a);` выведет число a типа `int` в поле ширины n .

Examples

Input

```
-5 4 5
```

Output

-5	-4	-3	-2	-1	0	1	2	3	
-5	25	20	15	10	5	0	-5	-10	-15
-4	20	16	12	8	4	0	-4	-8	-12
-3	15	12	9	6	3	0	-3	-6	-9
-2	10	8	6	4	2	0	-2	-4	-6
-1	5	4	3	2	1	0	-1	-2	-3
0	0	0	0	0	0	0	0	0	0
1	-5	-4	-3	-2	-1	0	1	2	3
2	-10	-8	-6	-4	-2	0	2	4	6
3	-15	-12	-9	-6	-3	0	3	6	9

Input

Output

Notes

Пояснение к примеру:

Вывод всех чисел должен осуществляться в поле шириной 5 символов, но между полями должен быть еще один пробел!!!

То есть первая строка: 5 пробелов, 1 пробел, поле шириной 5 символов (3 пробела и число -5 (занимает еще 2 символа)), 1 пробел, поле шириной 5 символов (3 пробела и число -4 (занимает 2 символа)) и так далее. После последнего символа строки сразу '`\n`'.

Submit a solution

Language: gcc - GNU C 7.5.0

File No file chosen

Send!

[Previous submissions of this problem](#)

PREVIOUS SUBMISSIONS OF THIS PROBLEM

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
139	2022/09/21 13:58:14	383	MZ4	gcc	OK	N/A	View	View
138	2022/09/21 13:57:04	372	MZ4	gcc	Wrong answer	5	View	View
119	2022/09/21 13:45:12	370	MZ4	gcc	Wrong answer	1	View	View
116	2022/09/21 13:41:04	357	MZ4	gcc	Wrong answer	1	View	View

[MZ0](#) [MZ1](#) [MZ2](#) [MZ3](#) [MZ4](#) [MZ5](#) [MZ6](#) [MZ7](#) [MZ8](#) [MZ9](#) [MZ10](#) [MZ11](#) [MZ12](#)[ejudge 3.8.0 #3 \(2020-09-10 08:07:26\).](#)

Copyright © 2000-2020 Alexander Chernov.



21:34:54 / 7 unread message(s) / RUNNING

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

Submit a solution for MZ5-Компактная упаковка даты

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Компактная дата (Упаковка)

Задана дата в виде YY MM DD, (Да, три числа через пробел!) где YY - две последние цифры года, MM-месяц, DD-день, например, 22 09 20. Требуется упаковать эту дату в 2 байта. Для кодирования дня (1 - 31) достаточно 5 бит, месяца (1-12) 4 бит, для двух последних цифр года (00-99) достаточно 7 бит. В итоге получится 16 бит или 2 байта. Таким образом требует использования битовые операции, упаковать дату в unsigned short.

Input format

%d %d %d (YY MM DD)

Output format

целое число типа unsigned short и бинарное представление компактной даты

Examples

Input

```
22 9 20
```

Output

```
0010110100110100  
11572
```

Notes

0010110100110100 <-> 0010110 1001 10100 <-> 22 9 20

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
282	2022/09/28 13:25:32	670	MZ5	gcc	OK	N/A	View	View
269	2022/09/28 13:09:57	560	MZ5	gcc	Rejected	N/A	View	View
267	2022/09/28 13:06:33	544	MZ5	gcc	Run-time error	1	View	View
266	2022/09/28 13:05:57	543	MZ5	gcc	Compilation error	N/A	View	View
265	2022/09/28 13:05:12	536	MZ5	gcc	Compilation error	N/A	View	View

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------





21:35:07 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ6-Распаковка даты

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Компактная дата (Распаковка)

Задано компактное представление даты в виде целого числа типа unsigned short. (более подробно смотрите задание B9) Требуется распаковать компактное представление даты записать его в виде трех целых чисел в формате: YYYY MM DD, где YYYY - год, MM - месяц, DD - день. Год выводить в формате 20XX, если XX меньше или равно 49 и в формате 19XX, если XX больше или равно 50.

Input format

целое число типа unsigned short

Output format

%d %d %d (YYYY MM DD)

Examples

Input

11572

Output

2022 9 20

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
375	2022/09/29 00:01:53	600	MZ6	gcc	OK	N/A	View	View
374	2022/09/28 23:57:01	629	MZ6	gcc	Wrong answer	4	View	View
313	2022/09/28 13:56:43	615	MZ6	gcc	Wrong answer	4	View	View
311	2022/09/28 13:55:05	620	MZ6	gcc	Compilation error	N/A	View	View

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12



21:35:21 / 7 unread message(s) / RUNNING

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-------------	------

Submit a solution for MZ7-Бит-реверс

Time limit: 1 s**Real time limit:** 5 s**Memory limit:** 64M

Бит-реверс

Ввести число и вывести другое число, двоичное представление которого совпадает с двоичным представлением введённого, но записанного справа налево (учитываются только значащие биты)

Input format

беззнаковое целое число

Output format

беззнаковое целое число

Examples

Input

13 (битовое представление 1101)

Output

11 (битовое представление 1011)

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
378	2022/09/29 00:16:29	280	MZ7	gcc	OK	N/A	View	View

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-------------	------

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

Copyright © 2000-2020 Alexander Chernov.



21:35:33 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ8-Маленькие, потом большие

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Маленькие, потом большие

Написать программу, которая считывает строку, состоящую из английских букв произвольной длины. Память под строку выделять динамически. Размер строки заранее неизвестен. Конец строки символ перевода строки или EOF. Полученную строку преобразовать таким образом, чтобы все прописные (маленькие) буквы в том порядке как они стоят оказались впереди заглавных (больших) букв. Оформить преобразование строки в виде отдельной функции. Запрещено использовать любую дополнительную память, кроме выделенной п строку.

Input format

строка из букв

Output format

строка из букв

Examples

Input

xMObRdpSjIceQfMx

Output

xbdpjcefxMORSIQM

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
900	2022/11/01 22:43:22	1597	MZ8	gcc	OK	N/A	View	View
870	2022/10/26 20:34:22	1572	MZ8	gcc	Rejected	N/A	View	View
869	2022/10/26 20:16:39	1588	MZ8	gcc	Run-time error	6	View	View
688	2022/10/19 15:54:33	1320	MZ8	gcc	Rejected	N/A	View	View
686	2022/10/19 15:53:06	1588	MZ8	gcc	Run-time error	6	View	View
681	2022/10/19 15:44:05	1576	MZ8	gcc	Run-time error	5	View	View
679	2022/10/19 15:36:43	1592	MZ8	gcc	Run-time error	5	View	View
678	2022/10/19 15:32:45	1564	MZ8	gcc	Run-time error	5	View	View
608	2022/10/13 18:55:20	1401	MZ8	gcc	Rejected	N/A	View	View
492	2022/10/06 16:47:52	1319	MZ8	gcc	Rejected	N/A	View	View
395	2022/10/05 13:10:29	1472	MZ8	gcc	Rejected	N/A	View	View
393	2022/10/05 13:07:27	1436	MZ8	gcc	Ignored	N/A	View	N/A

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

[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](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:35:47 / 7 unread message(s) / RUNNING

MZ0	MZ1	MZ2	MZ3	MZ4	MZ5	MZ6	MZ7	MZ8	MZ9	MZ10	MZ11	MZ12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

Submit a solution for MZ9-Цензура

Time limit: 2 s**Real time limit:** 5 s**Memory limit:** 64M

Цензура

Через параметры командной строки передаются нецензурные слова (последовательности маленьких латинских букв). На стандартный ввод подается текст содержащий слова (последовательности маленьких и больших латинских букв, разделенные пробелами и символами перевода строки). Вывести «отцензуренный» текст – текст, содержащий слова в исходной последовательности, но без нецензурных. Нецензурным также считается слово, написанное с Большой буквы или БОЛЬШИМИ буквами целиком.

Замечание: да, слова написанные большими и маленькими буквами вРАЗноБой – это обход цензуры!

Читать обязательно всю строку целиком (нельзя читать и выводить по слову), допустимо читать и выводить по одной строке, для простоты можно считать, что строка содержит не более 1000 символов. Но нельзя использовать дополнительную память, то есть удаляя слова нужно в этой же строке.

Input format

Еще раз: нецензурные слова передаются через параметры командной строки, текст передается через стандартный ввод.

Examples

Input

```
Command line: dolor vivamus elit
Stdin: Lorem ipsum dolor sit amet consectetur adipiscing ELIT Etiam condimentum vel est elit nec dolor posuere Vivamus tincidunt pretium risus
```

Output

```
Lorem ipsum sit amet consectetur adipiscing Etiam condimentum vel est nec dolor posuere tincidunt pretium risus
```

Notes

Слово, в котором нецензурное слово является его частью, нецензурным не считается и удалять его не надо!

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
664	2022/10/19 13:52:55	2184	MZ9	gcc	OK	N/A	View	View
661	2022/10/19 13:49:50	2170	MZ9	gcc	Ignored	N/A	View	N/A
606	2022/10/13 18:47:21	2134	MZ9	gcc	Rejected	N/A	View	View
592	2022/10/13 01:03:33	1964	MZ9	gcc	Wrong answer	2	View	View
591	2022/10/13 01:01:08	1944	MZ9	gcc	Wrong answer	2	View	View
582	2022/10/12 22:47:46	1858	MZ9	gcc	Wrong answer	2	View	View
580	2022/10/12 22:27:34	1836	MZ9	gcc	Wrong answer	2	View	View
579	2022/10/12 22:23:05	1831	MZ9	gcc	Wrong answer	2	View	View
577	2022/10/12 21:56:07	2609	MZ9	gcc	Wrong answer	2	View	View
576	2022/10/12 21:49:02	2584	MZ9	gcc	Wrong answer	2	View	View

572	2022/10/12 20:57:36	2296	MZ9	gcc	Run-time error	1	View	View
571	2022/10/12 20:55:15	2330	MZ9	gcc	Compilation error	N/A	View	View

[MZ0](#) [MZ1](#) [MZ2](#) [MZ3](#) [MZ4](#) [MZ5](#) [MZ6](#) [MZ7](#) [MZ8](#) [MZ9](#) [MZ10](#) [MZ11](#) [MZ12](#)

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

Copyright © 2000-2020 Alexander Chernov.



21:35:59 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ10-Работа с файлом

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Количество символов, слов, строк в файле

Реализовать программу, подсчитывающую количество строк, слов и символов в текстовом файле. Разделителями считать пробел. Символом является любой символ отличный от символа конца файла ('\n' тоже символ), словом является непустая последовательность символов между разделителями или между разделителем и символом конца строки или между началом файла и первым разделителем, строка - последовательность символов (возможно пустая!) заканчивающаяся символом конца строки

Результат три числа: количество строк, количество слов, количество символов разделенные пробелом записываются в другой файл. Имена файлов передаются через параметры командной строки. Первым идет имя файла для чтения, вторым имя файла для записи результата. Предусмотреть возможные нештатные ситуации (не смог открыться файл, неправильные параметры командной строки) во всех таких случаях вернуть код завершения 1. При возникновении нештатной ситуации на экран или в файл ничего не выводить.

Input format

File: последовательность текстовых символов

Command line: имя файла для чтения имя файла для записи

Output format

File: три целых числа

Examples

Input

```
mama mila ramu
papa mil divan
```

Output

```
2 6 30
```

Notes

В командной строке вводиться ./a.out input.txt output.txt

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
868	2022/10/26 20:11:56	950	MZ10	gcc	OK	N/A	View	View
657	2022/10/19 13:40:04	906	MZ10	gcc	Rejected	N/A	View	View
643	2022/10/19 13:12:21	802	MZ10	gcc	Wrong answer	4	View	View
639	2022/10/19 13:07:50	780	MZ10	gcc	Wrong answer	4	View	View
637	2022/10/19 13:05:47	751	MZ10	gcc	Compilation error	N/A	View	View

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

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

Copyright © 2000-2020 Alexander Chernov.



21:36:13 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ11-Удалить элемент

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Удалить элемент из списка

Написать функцию удаления элемента из списка в случае, если слово, которое храниться в элементе совпадает со словом заданным в аргументе функции. Прототип функции:

```
struct Inode* delete_element(struct Inode *head, char *str),
```

где head указатель на первый элемент списка, str - указатель на слово. Функция возвращает указатель на первый элемент списка или NULL. В случае если элементов удовлетворяющим условию удаления несколько удалить только первый элемент. Список задается считыванием слов из входного потока и использованием функции добавления в конец. (для всех элементов, кроме первого) Первым словом во входном потоке задается слово, которое будет использоваться для удаления элемента из списка. Второе и следующие слова относятся к списку. Распечатать полученный в результате удаления элемента список. Печать начинать с первого элемента списка.

Input format

Слова, состоящие из английских букв разделенные пробелом, каждое слово не более 20 символов. Количество слов неограничено.

Output format

Слова, состоящие из английских букв разделенные пробелом или слово "empty" если список пустой

Examples

Input

```
dear  
Hello my dear world
```

Output

```
Hello my world
```

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
963	2022/11/09 20:54:42	2559	MZ11	gcc	Rejected	N/A	View	View
915	2022/11/03 19:35:19	2511	MZ11	gcc	Rejected	N/A	View	View
866	2022/10/26 20:08:23	2489	MZ11	gcc	Rejected	N/A	View	View
865	2022/10/26 20:06:32	2490	MZ11	gcc	Compilation error	N/A	View	View
737	2022/10/20 12:39:52	2352	MZ11	gcc	Rejected	N/A	View	View
736	2022/10/20 12:36:32	2356	MZ11	gcc	Compilation error	N/A	View	View
735	2022/10/20 02:41:01	2139	MZ11	gcc	Run-time error	8	View	View
734	2022/10/20 02:33:55	2061	MZ11	gcc	Run-time error	8	View	View
733	2022/10/20 02:31:17	2042	MZ11	gcc	Wrong answer	7	View	View

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

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

Copyright © 2000-2020 Alexander Chernov.



21:36:28 / 7 unread message(s) / RUNNING

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

Submit a solution for MZ12-Два минимума

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Два минимума

Задан узел дерева

```
struct tnode{  
    int key;  
    struct tnode *left, *right;  
};
```

На стандартном вводе задана последовательность целых чисел, используя эту последовательность построить бинарное дерево так, что узел слева всегда имеет значение ключа меньше, а узел справа всегда имеет значение ключа большее, чем ключ текущего узла. Повторяющиеся элементы последовательности игнорировать. Используя построенное дерево написать функцию, которая печатает в стандартный вывод два различных наименьших ключа двоичного дерева поиска (то есть самый маленький ключ k_min и наименьший среди всех ключей, кроме k_min). Считать, что в дереве поиска содержится не менее двух вершин). В функции непосредственное сравнение ключей выполнять запрещено!

Input format

Последовательность целых чисел, минимум два числа

Output format

Два целых числа в порядке возрастания

Examples

Input

10 8 15 5 3 6 12 16

Output

3 5

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
964	2022/11/09 20:58:18	2339	MZ12	gcc	OK	N/A	View	View
914	2022/11/03 19:30:51	2694	MZ12	gcc	Rejected	N/A	View	View
875	2022/10/27 00:26:27	2657	MZ12	gcc	Rejected	N/A	View	View
864	2022/10/26 20:00:42	2653	MZ12	gcc	Ignored	N/A	View	N/A
857	2022/10/26 19:05:29	2629	MZ12	gcc	Wrong answer	7	View	View
855	2022/10/26 18:27:04	1829	MZ12	gcc	Wrong answer	3	View	View

MZ0 MZ1 MZ2 MZ3 MZ4 MZ5 MZ6 MZ7 MZ8 MZ9 MZ10 MZ11 MZ12

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

Copyright © 2000-2020 Alexander Chernov.



21:36:51 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-20-Обрезка файла

Time limit: 1 s**Real time limit:** 5 s**Memory limit:** 64M

Обрезка строк

Написать программу, которая копирует из текстового файла не более 50 символов каждой строки. То есть если в строке меньше и равно 50 символов, строка копируется целиком. (символ '\n' в эти 50 символов не входит), если строка больше 50 символов, копируются только первые 50, остальные игнорируются. Результат записать в другой файл. Имена файлов передаются через параметр командной строки.

Для копирования использовать низкоуровневые функции для работы с файлами (open, read, write, close)

В случае возникновения любых нештатных ситуаций в поток ошибок вывести "Error".

Строки в текстовом файле это последовательность ASCII символов заканчивающаяся символом перевода строки '\n'.

Реализовать копирование блоками по N байт, где N некоторая константа. Допустимо читать из файла блоком одного размера, записывать блоком другого размера, но недопустимо, чтобы размер блока равнялся 1 либо был кратен 50 или его делителям

Input format

Текстовый файл

В командной строке имена файлов.

Output format

Текстовый файл

Examples

Input

```
This is a long string with length more than 50 characters.  
But this string is short.  
And this string is last. Its length more than 50 characters.
```

Output

```
This is a long string with length more than 50 cha  
But this string is short.  
And this string is last. Its length more than 50 c
```

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
165	2022/11/16 20:23:47	2733	MZ-20	gcc	OK	N/A	View	View
85	2022/11/09 22:54:44	2641	MZ-20	gcc	Rejected	N/A	View	View
84	2022/11/09 22:52:21	2639	MZ-20	gcc	Wrong answer	2	View	View
29	2022/11/03 19:23:38	1379	MZ-20	gcc	Rejected	N/A	View	View
28	2022/11/03 19:20:38	1545	MZ-20	gcc	Ignored	N/A	View	N/A

id	date	user	language	status	score	view	edit
27	2022/11/03 19:04:24	1515	MZ-20	gcc	Wrong answer	3	View
13	2022/11/02 13:55:52	1375	MZ-20	gcc	Wrong answer	2	View
10	2022/11/02 13:38:21	1303	MZ-20	gcc	Wrong answer	1	View

[MZ-20](#)

[MZ-21](#)

[MZ-22](#)

[MZ-23](#)

[MZ-24](#)

[MZ-25](#)

[MZ-26](#)

[MZ-27](#)

[MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217 2022]: Submit a solution

[Settings](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:37:03 / 2 unread message(s) / RUNNING

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)

Submit a solution for MZ-21-В порядке возрастания

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

В порядке возрастания

В бинарном файле лежат целые знаковые числа типа int. Написать программу, которая запишет в другой файл эти числа в бинарном виде в порядке возрастания. Запрещено хранить одновременно больше двух чисел из файла. Имена файлов передаются через параметры командной строки. Разрешается изменение входного файла. В случае, если данные входящем (исходном) файле заданы некорректно, (то есть количество байт в файле не кратно sizeof(int)) выходной файл (результат) оставить пустым.

Input format

файл с бинарными знаковыми целыми числами типа int

Output format

файл с бинарными знаковыми целыми числами типа int

Examples

Input in input.bin

```
00000000 03 00 00 00 0a 00 00 00 02 00 00 00 0c 00 00 00 | ..... |
00000010 04 00 00 00 09 00 00 00 07 00 00 00 05 00 00 00 | ..... |
00000020
```

Output in output.bin

```
00000000 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 | ..... |
00000010 07 00 00 00 09 00 00 00 0a 00 00 00 0c 00 00 00 | ..... |
00000020
```

Notes

Пример приведен для случая, если командная строка имеет вид: ./a.out input.bin output.bin

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
91	2022/11/10 00:40:16	1233	MZ-21	gcc	OK	N/A	View	View
90	2022/11/10 00:38:22	1217	MZ-21	gcc	Presentation error	1	View	View
89	2022/11/10 00:31:56	1378	MZ-21	gcc	Presentation error	1	View	View
88	2022/11/10 00:27:07	1288	MZ-21	gcc	Presentation error	1	View	View

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)



21:37:16 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-22-Отец и сыновья

Time limit: 1 s**Real time limit:** 5 s**Memory limit:** 64M

Отец и сыновья

На стандартном потоке ввода задается целое число N (N > 0). Это число читает процесс отец. Требуется создать N процессов-сыновей.

Каждый процесс-сын выводит на **стандартный поток вывода** строку son n, где n порядковый номер этого сына. В конце строки выводит \n. Для вывода можно использовать средства высокого уровня,(printf) но помните, что printf использует буффер, поэтому не забывайте е сбрасывать. (функция fflush(stdout))

Кроме того каждый из процессов-сыновей выводит в **стандартный поток ошибок** свой PID и PPID в следующем формате: "PID = %d PPID = %d\n" и тоже не забываем про сброс буфера.

Процесс-родитель должен завершаться самым последним из всех процессов.

Input format

Целое неотрицательное число N

Output format

В stdout вывод будет:

```
son 1
son 2
...
son N
```

а в stderr будет:

```
"PID = %d PPID = %d\n"
"PID = %d PPID = %d\n"
...
"PID = %d PPID = %d\n"
```

Examples

Input

```
5
```

Output

вывод в stdout будет:

```
son 1
son 2
son 3
son 4
son 5
```

вывод в stderr будет:

```
PID = 4856 PPID = 4855
PID = 4857 PPID = 4855
PID = 4858 PPID = 4855
PID = 4859 PPID = 4855
PID = 4860 PPID = 4855
```

Notes

Обратите внимание, что PID и PPID на вашем компьютере естественно будет другим.

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
123	2022/11/16 13:01:11	463	MZ-22	gcc	OK	N/A	View	View
121	2022/11/16 12:58:58	477	MZ-22	gcc	Wrong answer	1	View	View

[MZ-20](#)

[MZ-21](#)

[MZ-22](#)

[MZ-23](#)

[MZ-24](#)

[MZ-25](#)

[MZ-26](#)

[MZ-27](#)

[MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.



21:37:26 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-23-Процесс и его потомки

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Процесс и его потомки

На стандартном потоке ввода задается целое число N ($N > 0$). Требуется создать N процессов по следующей схеме: процесс-родитель создает первый процесс (процесс-сын), процесс-сын создает второй процесс своей сыновий процесс (процесс-внук), процесс-внук создает третий процесс - своего сына (процесс-правнук) и так далее.

Процесс-родитель создает переменную pid, в которую записывает свой PID. Переменная pid создается в начале программы, до первого fork(). Дальше каждый процесс потомок выводит в **стандартный поток вывода** 2 числа. Первое число getpid()-pid(), второе число getpid()-pid. После этого выводится символ перевода строки.

Для вывода можно использовать средства высокого уровня. Вывод всегда происходит до порождения очередного процесса. Не забываем про сброс буфера. (функция fflush(stdout))

Кроме того каждый из процессов-потомков выводит в **стандартный поток ошибок** свой PID и PPID в следующем формате: "PID = %d PPID = %d\n" и тоже не забываем про сброс буфера.

Процесс-родитель каждого из процессов должен завершаться после завершения работы своего потомка.

Input format

Целое положительное число N

Output format

В stdout вывод будет:

```
1 1  
1 2  
1 3  
1 4  
...  
1 N
```

а в stderr будет:

```
"PID = %d PPID = %d\n"  
"PID = %d PPID = %d\n"  
..."  
"PID = %d PPID = %d\n"
```

Examples

Input

```
5
```

Output

вывод в поток stdout будет:

```
1 1  
1 2  
1 3  
1 4  
1 5
```

вывод в поток stderr будет:

```
PID = 3756 PPID = 3755  
PID = 3757 PPID = 3756  
PID = 3758 PPID = 3757  
PID = 3759 PPID = 3758  
PID = 3760 PPID = 3759
```

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
184	2022/11/18 17:55:19	562	MZ-23	gcc	OK	N/A	View	View
183	2022/11/18 17:54:12	562	MZ-23	gcc	Wrong answer	1	View	View
182	2022/11/18 17:53:24	667	MZ-23	gcc	Ignored	N/A	View	N/A
181	2022/11/18 17:48:27	667	MZ-23	gcc	Wrong answer	1	View	View
173	2022/11/17 00:01:01	578	MZ-23	gcc	Wrong answer	1	View	View
172	2022/11/16 22:28:32	793	MZ-23	gcc	Wrong answer	1	View	View
169	2022/11/16 21:50:49	789	MZ-23	gcc	Wrong answer	1	View	View
168	2022/11/16 21:42:14	1002	MZ-23	gcc	Wrong answer	1	View	View

[MZ-20](#)

[MZ-21](#)

[MZ-22](#)

[MZ-23](#)

[MZ-24](#)

[MZ-25](#)

[MZ-26](#)

[MZ-27](#)

[MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.



21:37:36 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-24-Двойной конвейер

Time limit: 1 s**Real time limit:** 5 s**Memory limit:** 64M

Двойной конвейер

Реализовать команду Shell с двумя конвейерами: pr1 arg11 ... arg1N \| pr2 arg12 ... arg2K \| pr3 arg1 ... arg3M, где pr1, pr2, pr3 имеи исполняемых файлов, arg11,...arg1N, arg21,...arg2K, arg31,..,arg3M - параметры (ключи) исполняемых файлов (могут отсутствовать)

Имена исполняемых файлов и их аргументов передаются через параметры командной строки в порядке их использования использованием "|". Обратите внимание, что выше использован символ '\|', а не '|'. Сделано это для того, чтобы экранировать символ '|', чтобы Shell не воспринимала этот символ в соответствии с его служебным значением, а передавала в программу как символ '|'.

Input format

Командная строка

Output format

Результат работы командной строки

Examples

Input

```
./a.out ls -l \| grep -v rwx \| wc
```

Output

Результат должен быть равен результату работы команды Shell:
ls -l | grep -v rwx | wc

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
486	2022/12/21 14:47:21	2306	MZ-24	gcc	OK	N/A	View	View
410	2022/12/14 16:26:00	2631	MZ-24	gcc	Rejected	N/A	View	View
409	2022/12/14 16:25:09	2615	MZ-24	gcc	Synchronization error	1	View	View
408	2022/12/14 16:15:19	2319	MZ-24	gcc	Synchronization error	1	View	View
407	2022/12/14 16:14:19	2338	MZ-24	gcc	Wall time-limit exceeded	1	View	View
406	2022/12/14 15:41:11	2362	MZ-24	gcc	Wall time-limit exceeded	1	View	View
358	2022/12/10 12:49:16	2385	MZ-24	gcc	Rejected	N/A	View	View
280	2022/11/30 13:30:50	2382	MZ-24	gcc	Rejected	N/A	View	View
187	2022/11/18 21:36:43	2310	MZ-24	gcc	Rejected	N/A	View	View
185	2022/11/18 21:32:18	2262	MZ-24	gcc	Ignored	N/A	View	N/A

MZ-20 MZ-21 MZ-22 MZ-23 MZ-24 MZ-25 MZ-26 MZ-27 MZ-28

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

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217 2022]: Submit a solution

[Settings](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:37:47 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-25-Пинг-понг 3-х процессов

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Пинг-понг 3 процесса (сигналы)

В программе создаются процесс-отец, процесс-сын1 и процесс-сын2. Необходимо реализовать игру пинг-понг между этими тремя процессами используя ОДИН неименованный канал синхронизация через сигналы. Начинает игру процесс-отец. Он читает со стандартного ввода два целых числа типа int и через канал посылает их процессу-сыну2. Процесс-сын1 печатает на стандартный вывод значение полученных чисел, уменьшает первое число на 1, а второе увеличивает на 1 и посылает процессу-сыну1, процесс-сын1 также печатает значение двух чисел, уменьшает первое на 1, а второе увеличивает на 1 и посылает процессу-отцу, процесс-отец печатает числа и уменьшает первое на 1, а второе увеличивает на 1, посылает процессу-сыну2 и так далее. Игра пинг-понг останавливается, когда первое число становится меньше второго числа. После буферизированной печати не забываем ставить flush(stdout)

Input format

Два целых чисел типа int

Output format

Каждый процесс печатает строчку, в которой указывает кто он "father" или "son1" или "son2" и через пробел текущие значения чисел.

Examples

Input

27 1

Output

```
son2 27 1
son1 26 2
father 25 3
son2 24 4
son1 23 5
father 22 6
son2 21 7
son1 20 8
father 19 9
son2 18 10
son1 17 11
father 16 12
son2 15 13
son1 14 14
```

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
329	2022/12/06 00:45:47	1848	MZ-25	gcc	Ignored	N/A	View	N/A
328	2022/12/06 00:44:19	1848	MZ-25	gcc	Wrong answer	1	View	View
278	2022/11/30 13:27:55	2311	MZ-25	gcc	OK	N/A	View	View
259	2022/11/25 12:01:49	2311	MZ-25	gcc	Wrong answer	2	View	View
257	2022/11/24 22:27:43	2014	MZ-25	gcc	Wrong answer	2	View	View
256	2022/11/24 22:20:16	2011	MZ-25	gcc	Wrong answer	2	View	View

ID	Date	Year	MZ	Compiler	Wrong answer	Score	View	View
255	2022/11/24 22:15:48	2022	MZ-25	gcc	Wrong answer	2	View	View
254	2022/11/24 22:14:10	2018	MZ-25	gcc	Compilation error	N/A	View	View

[MZ-20](#) [MZ-21](#) [MZ-22](#) [MZ-23](#) [MZ-24](#) [MZ-25](#) [MZ-26](#) [MZ-27](#) [MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.



21:37:58 / 2 unread message(s) / RUNNING

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

Submit a solution for MZ-26-Пинг-понг процессов на семафорах

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Пинг-понг 3 процессов (семафоры)

В программе создаются процесс-отец, процесс-сын1 и процесс-сын2. Необходимо реализовать игру пинг-понг между этими тремя процессами используя ОДИН неименованный канал синхронизации через **семафоры**. Начинает игру процесс-отец. Он читает со стандартного потока два числа и через канал посыпает их процессу-сын1. Процесс-сын1 уменьшает первое число и увеличивает второе число на 1 и посыпает процессу-отцу, процесс-отец также уменьшает на 1 первое и увеличивает на 1 второе число и посыпает процессу-сын2, процессу-сын2 выполняет с числами аналогичные действия и посыпает процессу-отцу. Процесс-отец также выполняет с числами аналогичные действия и посыпает переменную процессу-сын1. Игра пинг-понг останавливается, когда второе число становится больше первого. Каждый процесс печатает полученные переменные начиная с процесса сына. Для печати можно использовать высокоровневые функции печати, но после нужно обязательно очистить буфер (fflush(stdout)). Каждый процесс печатает свое имя "father", "son1" или "son2" и полученные значения переменных.(до изменения их значений)

Input format

Два целых числа типа int

Output format

%s %d %d

Examples

Input

23 1

Output

```
son1 23 1
father 22 2
son2 21 3
father 20 4
son1 19 5
father 18 6
son2 17 7
father 16 8
son1 15 9
father 14 10
son2 13 11
father 12 12
```

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
483	2022/12/20 20:45:37	2253	MZ-26	gcc	Wrong answer	1	View	View
482	2022/12/20 19:39:04	2758	MZ-26	gcc	OK	N/A	View	View
353	2022/12/08 19:40:03	2597	MZ-26	gcc	Rejected	N/A	View	View
331	2022/12/06 01:36:56	2371	MZ-26	gcc	Rejected	N/A	View	View
330	2022/12/06 01:29:23	2371	MZ-26	gcc	Ignored	N/A	View	N/A
327	2022/12/06 00:31:40	2330	MZ-26	gcc	Wrong answer	1	View	View
326	2022/12/06 00:22:17	2312	MZ-26	gcc	Wrong answer	1	View	View

320	2022/12/06 00:21:13	2278	MZ-20	gcc	wrong answer	1	View	View
325	2022/12/06 00:18:55	2278	MZ-26	gcc	Compilation error	N/A	View	View
323	2022/12/06 00:18:08	2278	MZ-26	gcc	Compilation error	N/A	View	View

MZ-20

MZ-21

MZ-22

MZ-23

MZ-24

MZ-25

MZ-26

MZ-27

MZ-28

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

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217 2022]: Submit a solution

[Settings](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:38:11 / 2 unread message(s) / RUNNING

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)

Submit a solution for MZ-27-Обработка ip адресов

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Обработка ip адресов

Программа должна создать три дочерних процесса. Первый созданный процесс должен обрабатывать только IPv4-адреса класса A, второй процесс — только IPv4-адреса класса B, третий процесс — только IPv4 адреса класса C. Каждый из этих процессов извлекает из адреса номер сети и печатает класс сети (A, B или C) и номер сети на стандартный поток вывода в шестнадцатеричном виде. Процесс-родитель считывает со стандартного потока ввода IP-адреса, записанные в десятично-точечной нотации, и передает их на обработку соответствующему процессу-сыну. Дочерние процессы выводят класс и номер сети строго в той очередности, в которой получают адреса от родителя. Для передачи данных синхронизации работы должна использоваться очередь сообщений. Когда отец достигает конца стандартного потока ввода сначала должны завершиться процессы-сыновья, зато процесс-отец. Созданные объекты IPC должны быть уничтожены.

Input format

четыре целых числа (от 0 до 255) разделенных точками

Output format

Имя класса (A, B, C) и номер сети в шестнадцатеричном виде

Examples

Input

```
4.57.255.36
130.44.131.158
200.51.238.160
```

Output

```
A 4
B 22c
C 833ee
```

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
491	2022/12/21 14:59:51	2450	MZ-27	gcc	Wrong answer	1	View	View
490	2022/12/21 14:57:39	2449	MZ-27	gcc	Wrong answer	1	View	View
412	2022/12/14 16:28:54	2653	MZ-27	gcc	Rejected	N/A	View	View
411	2022/12/14 16:27:47	2393	MZ-27	gcc	Wrong answer	1	View	View
362	2022/12/11 19:03:57	2382	MZ-27	gcc	Wrong answer	1	View	View
361	2022/12/11 18:57:18	2643	MZ-27	gcc	Rejected	N/A	View	View
360	2022/12/11 18:40:19	2250	MZ-27	gcc	Wrong answer	1	View	View
359	2022/12/11 18:39:13	2202	MZ-27	gcc	Synchronization error	1	View	View

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.



Задворный Даниэль [MSU CMC 217 2022]: Submit a solution

[Settings](#)[Info](#)[Summary](#)[Submissions](#)[Standings](#)[Submit clar](#)[Clars](#)[Logout \[cmc_217_2022_07\]](#)

21:38:22 / 2 unread message(s) / RUNNING

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)

Submit a solution for MZ-28-Собрать строку

Time limit: 1 s

Real time limit: 5 s

Memory limit: 64M

Очередь сообщений

В программе участвуют процесс-отец и 3 процесса-сына. Отец считывает со стандартного ввода целое число n и создает разделяемую память на $n+1$ байт. Далее создает пустую строку и записывает ее в разделяемую память. Каждый процесс-сын по очереди (сначала 1-й сын, потом 2-й, потом 3-й) добавляет ровно 1 символ к строке. При этом 1-й сын в строке добавляет маленькие английские буквы в порядке английского алфавита (a,b,c,...) 2-й процесс-сын добавляет цифры начиная с нуля (0,1,2,...). 3-й процесс-сын добавляет большие английские буквы в порядке английского алфавита (A,B,C,...). Процессы-сыновья обмениваются формируемой строкой, используя разделяемую память, синхронизация, используя семафоры. Если у процесса заканчиваются буквы или цифры, он начинает их добавлять начиная опять с первой буквы или цифры. После того как сформируется строка длины n (если длина не входит в пределы) Процесс-отец распечатывает полученную строку, дожидается всех сыновей, удаляет разделяемую память и массив семафоров и завершает программу.

Input format

Целое положительное число не больше 255

Output format

строка состоящая из цифр, английских больших и маленьких букв.

Examples

Input

10

Output

a0Ab1Bc2Cd

Input

50

Output

a0Ab1Bc2Cd3De4Ef5Fg6Gh7Hi8Ij9Ik0Kl1Lm2Mn3No4Op5Pq6

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
488	2022/12/21 14:53:54	3050	MZ-28	gcc	OK	N/A	View	View
487	2022/12/21 14:52:27	2981	MZ-28	gcc	Ignored	N/A	View	N/A
402	2022/12/14 13:49:53	2655	MZ-28	gcc	Rejected	N/A	View	View
399	2022/12/14 13:43:38	2623	MZ-28	gcc	Wrong answer	7	View	View

[MZ-20](#)[MZ-21](#)[MZ-22](#)[MZ-23](#)[MZ-24](#)[MZ-25](#)[MZ-26](#)[MZ-27](#)[MZ-28](#)

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

Copyright © 2000-2020 Alexander Chernov.