



Reverse (output-only task)

TASK

Considera una maquina con dos operaciones Two-Operation Machine (abreviado TOM) con nueve registros, enumerados 1...9. Cada registro almacena un entero no negativo en el rango 0...1000. La maquina tiene dos operaciones:

$S \ i \ j$	Almacena el valor del registro i más uno en el registro j . Nota que i puede ser igual a j .
$P \ i$	Imprime el valor del registro i .

Un programa para TOM incluye un conjunto de valores iniciales para los registros y una secuencia de operaciones. Dado un entero N ($0 \leq N \leq 255$), genera un programa para TOM que imprima la secuencia decreciente de enteros $N, N-1, N-2, \dots, 0$. El número más grande de S -operaciones consecutivas debe ser tan pequeño como sea posible.

Ejemplo de un programa para TOM y su ejecución para $N=2$:

Operación	Nuevos valores de los registros									Valor impreso
	1	2	3	4	5	6	7	8	9	
Valores iniciales	0	2	0	0	0	0	0	0	0	
P 2	0	2	0	0	0	0	0	0	0	2
S 1 3	0	2	1	0	0	0	0	0	0	
P 3	0	2	1	0	0	0	0	0	0	1
P 1	0	2	1	0	0	0	0	0	0	0

Los archivos de entrada están numerados de 1 hasta 16 y están disponibles vía el servidor de la competición.

Entrada:

- La primera línea del fichero de entrada contiene K , un entero que especifica el número del fichero de entrada.
- La segunda línea del fichero de entrada contiene N .

Ejemplo de entrada:

1
2

Salida:

La primera línea de la salida debe ser el string "FILE reverse K ", donde K es el numero del fichero de entrada.



La segunda línea del fichero de salida nueve valores separados por un espacio representando los valores iniciales deseados para los registros, en orden (registro 1, después registro 2, etc.).

El resto del fichero de salida debe contener la lista ordenada de operaciones que deben desarrollarse, una por línea. Así, la tercera línea contiene la primera operación a desarrollar, y así sucesivamente. La última línea del fichero debe ser una que imprime 0. Cada línea debe contener una operación válida. Las instrucciones deben tener el formato mostrado en el ejemplo de salida.

Ejemplo de salida #1 (puntuación parcial):

```
FILE reverse 1
0 2 0 0 0 0 0 0 0
P 2
S 1 3
P 3
P 1
```

Ejemplo de salida #2 (puntuación plena):

```
FILE reverse 1
0 2 1 0 0 0 0 0 0
P 2
P 3
P 1
```

PUNTUACION

La puntuación de cada test se basará en la validez y la optimalidad del programa para TOM dado.

Validez: 20%

Un programa para TOM es correcto si no desarrolla más de 131 S-operaciones consecutivas y la secuencia de los valores impresos es correcta (contiene exactamente $N+1$ enteros, empezando en N y acabando en 0). Si alguna S-operación provoca el desbordamiento de un registro, el programa para TOM se considera incorrecto.

Optimalidad: 80%

La optimalidad de un programa para TOM se mide por el máximo número de S-operaciones consecutivas en el programa, que debe ser tan pequeño como sea posible. La puntuación se basa en la diferencia entre tu programa para TOM y el mejor programa para TOM conocido.