

Enseñando recursión

McFly, parece que vas haciendo avances. Gracias a los libros de la biblioteca a tus futuros padres les está picando el gusanillo de la informática; gracias a la ayuda que das a Doc para resolverles las dudas que le preguntan ya controlan los algoritmos iterativos; por último, gracias a vuestra visita al ENIAC de hace dos semanas, quieren tener más contacto con los ordenadores.

Tanto es así, que estás pensando en enseñarles tu CPC-6128 traído del futuro para que aprendan recursión con él. A Doc no le gusta mucho la idea, porque esa extraña máquina (“¡Santo cielo!” dijo la primera vez que la vió) va a levantar muchas sospechas, pero tú estás decidido.

Entre los discos que has encontrado en la mochila hay, de hecho, uno que al más puro estilo de “juego educativo” va proponiendo, de forma secuencial, problemas de programación de bucles (B) y de recursión (R).

Como una vez elegido el problema inicial (se puede empezar por cualquiera de ellos) hay que hacerlos todos seguidos, quieres saber cuántos podrán hacer teniendo en cuenta que:

- Quieres que hagan el máximo número de problemas posible.
- No quieres que hagan más de k problemas de bucles; al fin y al cabo los bucles ya parece que se los saben y no quieres que pierdan la motivación.

Entrada

La entrada estará compuesta por distintos casos de prueba, cada uno en una línea.

Cada línea empieza con un número k que indica el número máximo de ejercicios de bucles que tus padres están dispuestos a hacer. Le sigue una cadena con tantos caracteres como problemas hay en el disco que simboliza el tipo de problema. Una B denota un problema de bucles y una R un problema de recursión. En un disco siempre hay al menos un problema pero nunca entran más de 1.000.000.

Salida

Por cada caso de prueba se escribirá una línea con un número que indica el número máximo de problemas que podrán hacer tus padres.

Entrada de ejemplo

```
0 RRRRR
0 BRRBR
1 BRRBR
2 BRRBR
2 BBBB
2 RBBRB
```

Salida de ejemplo

```
5
2
4
5
2
4
```

Notas

Ten en cuenta que:

- Para que el ejercicio sea evaluado debe ser primero aceptado por el juez.

- Debes tener al menos la función siguiente:
`int maxProblemas(char v[], int n, int k);`¹ que devuelva el número máximo de problemas que podrán hacer manteniendo las restricciones del enunciado.
- Esta función debe venir acompañada de su especificación (precondición y postcondición), del invariante del bucle utilizado y de la cota.
- Debes también indicar en el código la complejidad del algoritmo.

Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2018/2019 (prof. Marco Antonio Gómez Martín). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.

¹El tipo del parámetro `v` puede ser `const string&`. En ese caso se puede omitir el parámetro `n`.