The Virtual Learning Environment for Computer Programming

Word Wrap X57785_es

El objetivo de la práctica es escribir un programa que formatea párrafos de manera óptima. Dado un valor $W \geq 0$ que es la anchura de la línea, y un párrafo P queremos determinar los lugares óptimos donde colocar los saltos de línea, de manera que todas tengan líneas de longitud lo más parecido posible a W. Consideramos que un párrafo P está formado por una secuencia de palabras separadas entre sí por un espacio en blanco. En la entrada podemos tener espacios en blanco antes de la primera palabra de un párrafo, después de la última palabra o espacios múltiples entre las palabras, por lo que un paso previo consistirá en eliminar todos los espacios en blanco innecesarios.

El proceso de optmización por lo tanto consiste en determinar qué espacios en blanco del párrafo P deben ser sustituídos por saltos de línea. Para una línea de longitud ℓ (sin contar el salto de línea del final), se define el*penalty* como el cuadrado de la diferencia entre ℓ y W:

penalty =
$$(\ell - W)^2$$
.

El penalty de un párrafo P formateado en líneas $L_1, L_2, ..., L_k$ es la suma de los penalties de las líneas. El objetivo será entonces dividir el párrafo P dado de tal modo que se minimice el penalty.

Las soluciones voraces no funcionan correctamente; hay que utilizar programación dinámica para resolver este problema.

Entrada

La entrada comienza con el valor *W*. A continuación viene una secuencia de párrafos (con espacios redundantes que deben ser eliminados); cada párrado viene separado del siguente por una o más líneas en blanco. Por ejemplo,

```
este____es_el_primer_parrafo,_que_consiste_en
varias
palabras,_teniendo_en_cuenta_que_los_signos
de_puntuacion_como_la_coma
se_consideran_parte_de_la_palabra.

por_ejemplo,_en
el
parrafo_anterior,_el
punto_final_es_parte
de_la
palabra_"palabra."

____este_tercer_parrafo____esta_separado
___del_anterior_por_varias_lineas_en_blanco
y_hay_espacios_redundantes_antes_de_la_primera_palabra,
separando_palabras_y_tambien_al_final
```

En el ejemplo los espacios en blanco se han marcado para hacerlos más fácilmente visibles.

Salida

La secuencia de párrafos formateados óptimamente, a continuación una línea que pone Penalty: y el valor del penalty correspondiente al párrafo. Cada párrafo viene separado del siguiente por una línea en blanco.

Observación

Vuestro programa deberá estar escrito en C++. Para la evaluación de la práctica se usarán juegos de pruebas públicos y privados, pero el *Jutge* sólo tendrá en cuenta los juegos de pruebas públicos para determinar si el programa pasa o no, y los resultados de los juegos de pruebas privados sólo se usarán para perfilar la nota de la práctica.

En el fichero entregado, el algoritmo deberá estar debidamente comentado. En particular debéis justificar su corrección y su eficiencia para formatear un solo párrafo en función del número n de palabras que constituyen el párrafo.

La fecha y hora límite de entrega se publicará en un aviso del Racó.

Ejemplo de entrada 1

```
2.0
el primer
parrafo
consiste
          en una
serie de lineas bastante
irregulares
    el segundo parrafo tampoco es moco
de
pavo, hay bastante texto que
esta formateado de manera irregular
pero
10
mejor
    todo es
   tercer parrafo
```

Ejemplo de salida 1

el primer parrafo consiste
en una serie de lineas
bastante irregulares
Penalty: 40

el segundo parrafo
tampoco es moco de
pavo, hay bastante texto
que esta formateado
de manera irregular
Penalty: 26

pero lo mejor de todo
es el tercer parrafo
Penalty: 1

Ejemplo de entrada 2

6 aaa bb cc ddddd

Ejemplo de salida 2

aaa bb
cc ddddd
Penalty: 4

Información del problema

Autor : Conrado Martinez Generación : 2013-03-06 00:18:26

© *Jutge.org*, 2006–2013. http://www.jutge.org