

Problem F. Enunciados difíciles

Time limit 2000 ms

Mem limit 262144 kB

El equipo docente del Taller de Programación se confundió e hizo una tarea muy difícil. Para que no vuelva a pasar, ocuparán su súper transformador de problemas difíciles.

Consideraremos que el enunciado de un problema es un string. El equipo considera que un problema es difícil si su enunciado contiene la subsecuencia **hard**. Una subsecuencia se obtiene borrando uno o más caracteres del string.

Por ejemplo, los enunciados **hard**, **hxayrzd**, **haaaaard** son difíciles, mientras que **har**, **hart**, **drah** no lo son.

El equipo docente no quiere que los enunciados sean difíciles. Para ello, pueden borrar algunos caracteres del enunciado para que deje de serlo. Sin embargo, algunas partes del enunciado pueden ser cruciales para entenderlo. Inicialmente, la ambigüedad del enunciado es 0, y remover el i -ésimo carácter incrementa la ambigüedad en a_i .

El equipo docente quiere ocupar el súper transformador de problemas difíciles para que escoja los caracteres a borrar. El problema es que nadie lo ha programado, así que decidimos que una buena tarea es que tú lo programes.

Calcula la mínima ambigüedad de enunciado que se puede obtener, removiendo algunos caracteres (posiblemente cero) de forma que el enunciado no sea difícil.

Entrada

La primera línea contiene un entero n ($1 \leq n \leq 10^5$), el largo del enunciado.

La segunda línea contiene un string s de largo n , consistiendo en letras minúsculas del alfabeto inglés, representando el enunciado.

La tercera línea contiene n enteros a_1, a_2, \dots, a_n ($1 \leq a_i \leq 998244353$).

Salida

Imprime la mínima ambigüedad posible después de eliminar algunos caracteres (posiblemente cero) de forma que el enunciado resultante no sea difícil.

Ejemplo 1

Entrada	Salida
6 hhardh 3 2 9 11 7 1	5

Ejemplo 2

Entrada	Salida
8 hhzarwde 3 2 6 9 4 8 7 1	4

Ejemplo 3

Entrada	Salida
6 hhaarr 1 2 3 4 5 6	0

Note

En el primer ejemplo, se eliminan los primeros dos caracteres resultando **ardh** .

En el segundo, se elimina el quinto caracter, resultando **hhzawde** .

En el tercero no hay necesidad de eliminar caracteres.