

# [ED238] Código Binário

Neste problema deverá submeter uma classe **ED238** contendo um programa completo para resolver o problema (ou seja, com o método main). Não será adicionado nenhum código base ao seu programa, pelo que tem de incluir na submissão quaisquer classes que sejam necessárias para além das classes do próprio Java.

**[PROBLEMAS PARA DOWNLOAD]** Para precaver uma possível intermitência na ligação de internet, podem e devem fazer download de todos os problemas em: [https://mooshak.dcc.fc.up.pt/~edados/teste\\_parte1/NUM\\_MECANOGRAFICO.zip](https://mooshak.dcc.fc.up.pt/~edados/teste_parte1/NUM_MECANOGRAFICO.zip) (onde NUM\_MECANOGRAFICO deve ser substituído pelo vosso número mecanográfico)

Imagine que tem uma sequência de 0's e 1's e que pode modificar no máximo  $K$  elementos da sequência (ou seja alterar um elemento, trocando de 0 para 1 ou vice-versa). Qual seria a maior subsequência contígua de 1's que conseguiria formar?

Imagine por exemplo que tem a seguinte sequência:

0001100111101011000010100000001110111

Se  $K=0$  (ou seja, não pode alterar nenhum elemento), então a maior subsequência que consegue formar tem tamanho 4:

0001100**1111**01011000010100000001110111

Se  $K=1$  (ou seja, pode alterar apenas um elemento), então a maior subsequência que consegue formar tem tamanho 7 (o elemento alterado está indicado a vermelho):

000110011110101100001010000000**1111111**

Se  $K=2$  (ou seja, pode alterar dois elementos), então a maior subsequência que consegue formar tem tamanho 9 (os elementos alterados estão indicados a vermelho):

0001100**11111111111**000010100000001110111

Se  $K=4$  (ou seja, pode alterar quatro elementos), então a maior subsequência que consegue formar tem tamanho 13 (os elementos alterados estão indicados a vermelho):

000**111111111111111**000010100000001110111

## O Problema

Dada uma sequência de  $N$  0's e 1's e  $Q$  possíveis inteiros  $K$ , a sua tarefa é descobrir, para cada  $K$ , qual a maior subsequência contígua de 1's que consegue obter substituindo no máximo  $K$  0's por 1's.

Uma solução "bruta" não passará no tempo limite (mas para um conjunto de testes valendo 30% dos pontos,  $N$  é tão pequeno que até algo pouco eficiente poderá passar). Para ter pontuação completa é necessário e expectável que faça uma solução linear para cada  $K$ , ou seja, a sua complexidade final deverá ser  **$O(Q * N)$**

## Input

A primeira linha contém um inteiro **N**, o tamanho da sequência a considerar ( $1 \leq N \leq 100\,000$ ). A segunda linha contém a sequência, formada por **N** zeros e uns.

A terceira linha contém um inteiro **Q**, a quantidade de K's a considerar ( $1 \leq Q \leq 100$ ). A quarta linha contém **Q** inteiros (separados por um espaço), os vários  $K$ 's a considerar ( $0 \leq K \leq N$ ).

## Output

O output deve conter **Q** linhas, uma para cada  $K$  (na mesma ordem do input), contendo cada uma das linhas um único inteiro indicando o tamanho da maior sequência de 1's que se consegue formar mudando no máximo o número correspondente de dígitos da sequência original, como atrás descrito.

## Exemplo de Input/Output

Input	Output
37	4
0001100111101011000010100000001110111	7
5	9
0 1 2 4 3	13
	10

Nota: o exemplo de input corresponde ao exemplo dado no enunciado: 4 é a resposta para  $K=0$ , 7 é a resposta para  $K=1$ , 9 é a resposta para  $K=2$  e 13 é a resposta para  $K=4$  e 10 é a resposta para  $K=3$ .

