

## Sfida all'ultimo pangramma (pangramma)

Mojito, con l'aiuto di *tutti* gli altri Jack Russell Terrier materani, ha appena finito di appendere un lunghissimo striscione nella sala cerimonie dell'IIS G.B. Pentasuglia in vista della premiazione delle Olimpiadi di Informatica. Mentre assapora il meritato riposo, Mojito nota che nella lunga lista di sponsor è possibile identificare molti **pangrammi**: frasi in cui vengono utilizzate tutte le lettere dell'alfabeto.



The quick brown fox jumps over the lazy dog  
The five boxing wizards jump quickly  
How quickly daft jumping zebras vex  
QUICK ZEPHYRS BLOW VEXING DAFT JIM  
SPHINX OF BLACK QUARTZ JUDGE MY VOW  
How razorback jumping frogs can level six piqued gymnasts

Figura 1: Una selezione di sei diversi pangrammi<sup>1</sup>

Ovviamente, non è difficile trovare pangrammi lunghi, mentre è molto più difficile trovarne di corti. I pangrammi più corti possibili sono detti **eteroletterali**, e usano ogni lettera *esattamente una volta*:

Mr Jock TV quiz PhD bags few lynx

Purtroppo, non è detto che ci sia un pangramma così corto nello striscione. In mancanza, Mojito si può accontentare di trovare i **pangrammi di lunghezza minima** tra quelli presenti nello striscione.

Ispirato dalla scoperta, Mojito decide di organizzare una sfida con i suoi aiutanti: a turno, ciascuno dei **46 337** Jack Russell Terrier dovrà mostrare agli altri un pangramma minimo. Per farlo, dovrà evidenziare un'occorrenza di ogni lettera dell'alfabeto nello striscione, ovviamente in un modo diverso da quanto gli altri hanno già fatto in precedenza. Prima di iniziare la sfida, Mojito vuole capire in quanti modi diversi è possibile fare questa "evidenziazione"... altrimenti potrebbe rischiare di sfigurare!

Più precisamente, lo striscione riporta una stringa  $V$  di  $N$  caratteri, ciascuno preso da un alfabeto di  $K$  simboli.<sup>2</sup> I simboli sono rappresentati dai valori interi da 0 a  $K - 1$ . Una *evidenziazione* associa ad ogni simbolo  $j$  dell'alfabeto una posizione  $x_j$  nella stringa  $V$  dove compare tale simbolo, quindi tale che  $V[x_j] = j$ . Due evidenziazioni sono diverse se associano posizioni diverse  $x_j \neq x'_j$  ad almeno un simbolo  $j$ , anche se corrispondono allo stesso pangramma. Un'evidenziazione è *minima* se individua un pangramma minimo (cioè, se la massima differenza  $x_j - x_i$  tra le posizioni di due occorrenze è minima possibile). Mojito vuole contare quante evidenziazioni minime diverse esistono, **modulo 46 337**: aiutalo tu!

<sup>1</sup>I pangrammi, oltre a stuzzicare Mojito, trovano effettivamente un utilizzo nella vita reale. Ad esempio, in tipografia i pangrammi sono utili per testare la resa grafica dei caratteri di stampa. Inoltre, telegrafare un pangramma è un ottimo modo per chi vuole mettere alla prova la propria conoscenza del codice Morse. Negli anni '80, anche lo scrittore Umberto Eco era ossessionato dai pangrammi. Un pangramma eteroletterale in lingua italiana da lui proposto fu il seguente:

*"Tv? Quiz, BR, FLM, DC... Oh, spenga!"*

<sup>2</sup>Nello striscione non ci sono solo lettere: ci possono essere numeri, caratteri speciali, emoji, il simbolo di batman, ...

## Implementazione

✎ È importante usare sempre il modulo dopo una somma perché i valori intermedi generati possono richiedere più di 32 bit. Quando si cerca un risultato modulo  $M$ , suggeriamo di sfruttare il fatto che  $(A + B + C) \text{ modulo } M = (((A + B) \text{ modulo } M) + C) \text{ modulo } M$ . La stessa cosa vale per la moltiplicazione. In questo modo, i risultati intermedi restano sempre modulo  $M$  e possono essere mantenuti in una variabile intera.

Dovrai sottoporre un unico file, con estensione `.c` o `.cpp`.

📁 Tra gli allegati a questo task troverai dei template `pangramma.c` e `pangramma.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C	<code>int conta(int N, int K, int* V);</code>
C++	<code>int conta(int N, int K, vector&lt;int&gt;&amp; V);</code>

- L'intero  $N$  rappresenta la lunghezza dell'array  $V$ , ovvero il numero totale di lettere nello striscione.
- L'intero  $K$  rappresenta il numero di simboli diversi che sono presenti nell'alfabeto.
- L'array  $V$ , indicizzato da 0 a  $N - 1$ , indica il simbolo contenuto in ciascuna posizione.
- La funzione deve restituire il numero di evidenziazioni minime diverse modulo 46337.

Il grader chiamerà la funzione `conta` e ne stamperà il valore restituito sul file di output.

## Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che puoi usare per testare le tue soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama la funzione che devi implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da due righe, contenenti:

- Riga 1: due interi  $N$  e  $K$ .
- Riga 2:  $N$  interi  $V[i]$  per  $i = 0 \dots N - 1$ .

Il file di output è composto da un'unica riga, contenente il valore restituito dalla funzione `conta`.

## Assunzioni

- $1 \leq K \leq N \leq 1\,000\,000$ .
- $0 \leq V[i] \leq K - 1$  per  $i = 0 \dots N - 1$ .
- Tutti i simboli compaiono almeno una volta nella stringa.

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

- **Subtask 1** [ 0 punti]: Casi d'esempio.
- **Subtask 2** [ 5 punti]:  $K = 2$ .
- **Subtask 3** [ 7 punti]:  $N \leq 30$ .

- Subtask 4 [14 punti]:  $N \leq 300$ .
- Subtask 5 [13 punti]:  $N \leq 5000$ .
- Subtask 6 [12 punti]: Nello striscione c'è almeno un pangramma *eteroletterale*.
- Subtask 7 [11 punti]: Il pangramma minimo è lungo al massimo 60.
- Subtask 8 [10 punti]: Il numero di evidenziazioni minime è al massimo  $10^9$ .
- Subtask 9 [ 9 punti]:  $N \leq 500\,000$ .
- Subtask 10 [10 punti]: Nessuna limitazione specifica.
- Subtask 11 [ 7 punti]:  $\triangle$  Caso speciale:  $1\,000\,000 \leq K \leq N \leq 5\,000\,000$ .
- Subtask 12 [ 2 punti]:  $\triangle$  Caso speciale:  $5\,000\,000 \leq K \leq N \leq 9\,000\,000$ .

## Esempi di input/output

stdin	stdout
6 3 2 0 1 1 2 0	2
7 3 0 1 1 2 1 1 0	4
7 3 0 1 1 2 0 0 1	1
13 5 2 3 0 0 3 1 3 4 3 3 0 3 2	10

## Spiegazione

Nel **primo caso di esempio** sono presenti due pangrammi minimi, che sono anche eteroletterali:

$\boxed{2, 0, 1}$ , 1, 2, 0     2, 0, 1,  $\boxed{1, 2, 0}$

Ci sono anche altri pangrammi più lunghi, che però non vanno contati:

2,  $\boxed{0, 1, 1, 2}$ , 0

In totale, ci sono quindi **2** modi diversi per evidenziare un pangramma di lunghezza minima.

Nel **secondo caso di esempio**, sono di nuovo presenti due pangrammi minimi, anche se non sono eteroletterali, per un totale di **4** evidenziazioni minime:

$\boxed{0, 1, 1, 2}$ , 1, 1, 0     0, 1, 1,  $\boxed{2, 1, 1, 0}$   
 $\boxed{0, 1, 1, 2}$ , 1, 1, 0     0, 1, 1,  $\boxed{2, 1, 1, 0}$

Nel **terzo caso di esempio** c'è un unico pangramma eteroletterale, al centro, e quindi c'è una sola evidenziazione minima.

Nel **quarto caso di esempio** ci sono 2 pangrammi minimi, rispettivamente con 6 e con 4 evidenziazioni minime, per un totale di 10.