

Rank ed unrank di stringhe su alfabeto binario (rank_unrank_ABstrings)

Una stringa è una sequenza finita di caratteri presi da un qualche alfabeto di riferimento. Ad esempio, $s = ABBA$ è una stringa sull'alfabeto $\{A, B\}$. Indichiamo con $len(s)$ il numero di caratteri della stringa s , in questo caso 4. Ci concentreremo sulle sole stringhe sull'alfabeto $\{A, B\}$, non prevediamo altri caratteri. Ecco le $2^3 = 8$ stringhe di lunghezza 3 disposte in ordine lessicografico:

stringa 0: AAA	stringa 4: BAA
stringa 1: AAB	stringa 5: BAB
stringa 2: ABA	stringa 6: BBA
stringa 3: ABB	stringa 7: BBB

Ci servirebbe una tua implementazione della seguente funzione che riceve in ingresso una stringa s e ritorna la sua posizione (detta anche *rango*) nell'ordine delle $2^{len(s)}$ stringhe di lunghezza $len(s)$.

```
def ABstring2rank(s):  
    return 42
```

Pagherebbe inoltre una tua implementazione della seguente funzione che riceve in ingresso un numero naturale $length$ ed un numero naturale $r < 2^{length}$ e restituisce la r -esima stringa (sempre con riferimento all'ordine lessicografico) di lunghezza $length$.

```
def ABstring_of_len_and_rank(length, r):  
    return "ABBA"
```

Trovi un template della soluzione nel file `AB_strings_rank_and_unrank_template_sol.py`, dovrai solo risistemare l'implementazione delle due funzione richieste che al momento restituisce sempre la stessa risposta (anche un orologio rotto segna l'ora esatta due volte al giorno). Realizzarne correttamente anche una sola delle due già ti consegna dei punti.

Dati di input

Il vostro programma riceve in input o una stringa s sull'alfabeto $\{A, B\}$ oppure una coppia di numeri naturali $length$ ed r separati da uno spazio.

Dati di output

Nel primo caso (mi riferisco ai due casi di possibile input) il programma restituisce la posizione occupata da s tra le stringhe su $\{A, B\}$ di lunghezza $len(s)$ disposte in ordine lessicografico; si noti che la primissima di tali stringhe, quella composta di sole A , ha rango 0.

Nel secondo caso dobbiamo restituire la stringa di rango r tra le 2^{length} stringhe di lunghezza in $length$.

Esempi di input/output

input (da stdin)	output (su stdout)
ABAB	5
input (da stdin)	output (su stdout)
AAAA	0
input (da stdin)	output (su stdout)
BAAAAAAB	129
input (da stdin)	output (su stdout)
8 129	BAAAAAAB

Assunzioni

- tratteremo solo stringhe di lunghezza tra 1 e 100

Subtask

- **Subtask 1 [0 punti]:** gli esempi del testo.
- **Subtask 2 [10 punti]:** $\text{len}(s) \leq 1$, rank implementato correttamente.
- **Subtask 3 [10 punti]:** $\text{length} \leq 1$, unrank implementato correttamente.
- **Subtask 4 [10 punti]:** $\text{len}(s) \leq 2$, rank implementato correttamente.
- **Subtask 5 [10 punti]:** $\text{length} \leq 2$, unrank implementato correttamente.
- **Subtask 6 [10 punti]:** $\text{len}(s) \leq 3$, rank implementato correttamente.
- **Subtask 7 [10 punti]:** $\text{length} \leq 3$, unrank implementato correttamente.
- **Subtask 8 [10 punti]:** $\text{len}(s) \leq 10$, rank implementato correttamente.
- **Subtask 9 [10 punti]:** $\text{length} \leq 10$, unrank implementato correttamente.
- **Subtask 10 [10 punti]:** $\text{len}(s) \leq 100$, rank implementato correttamente.
- **Subtask 11 [10 punti]:** $\text{length} \leq 100$, unrank implementato correttamente.