



CHALLENGE “The Minion Game”

>> RETO ENERO <<





CHALLENGE “The Minion Game”

>> RETO ENERO <<





CHALLENGE “The Minion Game”

>> RETO ENERO <<





TEAM #11

>> Entusiastas de los datos <<





The Minion Game



Kevin y Stuart quieren jugar “**The Minion Game**”.

Reglas del juego:

- A ambos jugadores se les da la misma cadena.
- Ambos jugadores tienen que hacer subcadenas usando las letras de la cadena.

- > Stuart hacer palabras empezando por **consonantes**.
- > Kevin hacer palabras empezando por **vocales**.

El juego termina cuando ambos jugadores han hecho todas las subcadenas posibles.

Puntaje:

Un jugador obtiene punto por cada aparición de la subcadena de la cadena.





The Minion Game



Ejemplo:

String $S = \text{BANANA}$

- **KEVIN - VOCALES**
ANA aparece 2 veces en $S \Rightarrow 2$ pts



- **STUART - CONSONANTES**
NA aparece 2 veces en $S \Rightarrow 2$ pts



BANANA			
STUART 		KEVIN	
WORDS	SCORE	WORDS	SCORE
B	1	A	3
N	2	AN	2
BA	1	ANA	2
NA	2	ANAN	1
BAN	1	ANANA	1
NAN	1		
BANA	1		
NANA	1		
BANAN	1		
BANANA	1		
TOTAL	12	TOTAL	9



The Minion Game



✦ Consideración:

- La cadena S contendrá solo letras mayúsculas [A-Z]
- Las vocales solo se definen como AEIOU . En este problema, no se considera Y una vocal.
- Si el juego es un empate, imprima *Draw*.
- Imprimir al ganador en una línea con el sgte. Formato: **Nombre del ganador puntaje**

Entrada de muestra

BANANA

Salida de muestra

Stuart 12





Solución - Explicación



IDEA PLANTEADA

- Encontrar todas las subcadenas de una cadena
- Encontrar todas las subcadenas que empiecen con alguna vocal

PASO N° 1:

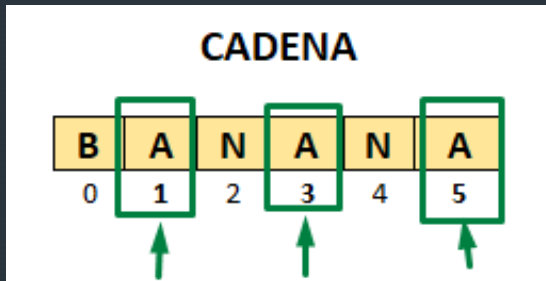
Se procedió a hallar la cantidad de subcadenas que se puede obtener de una cadena; se utilizó inducción matemática como se muestra a continuación:

CADENA	N° DE CARACTERES	SUBCADENAS	N° DE SUBCADENAS	
A	1	A	1	$= (1)(2)/2$
AB	2	A,B,AB	3	$= (2)(3)/2$
ABC	3	A,B,C,AB,BC	5	$= (3)(4)/2$
ABCD	4	A,B,C,D,AB,BC,CD,ABC,BCD,ABCD	10	$= (4)(5)/2$
ABCDE	5	A,B,C,D,E,AB,BC,CD,D,E,ABC,BCD,CDE,ABCD,BCDE,ABCDE	15	$= (5)(6)/2$
...				...
ABC...	N	...		$= (n)(n+1)/2$



PASO N° 2:

Se procedió a hallar la cantidad de subcadenas que se puede obtener de una cadena; se utilizó inducción matemática como se muestra a continuación:



SUMA = 0

# POS	SUBCADENA	# SUBCADENAS	LEN(CADENA) - POS
1	A AN ANA ANAN ANANA	5	= 6 - 1
3	A AN ANA	3	= 6 - 3
5	A A	1	= 6 - 5

SUMA = 5

SUMA = 8

SUMA = 9



Solución - Código



Concepto Previos

```
string = 'BANANA'

print('Pos','Caracter')

for index, character in enumerate(string):
    print(index, ' - ',character)
```

Pos	Caracter
0	- B
1	- A
2	- N
3	- A
4	- N
5	- A



Resolución completa

```
def minion_game(string):
    len_string = len(string)

    # Nro total de substrings de una cadena
    total_substrings = (len_string)*(len_string+1)/2

    suma = 0
    for index, character in enumerate(string):
        if character in 'AEIOU':
            cant_subcadenas = len_string - index
            suma = suma + cant_subcadenas

    # Nro total de substrings que empiezan con VOCAL
    total_substrings_vowels = suma

    # Nro total de substrings que empiezan con CONSONANTE
    total_substrings_consonants = int(total_substrings) - int(total_substrings_vowels)

    # Resultado:
    if total_substrings_vowels == total_substrings_consonants:
        print('Draw')
    elif total_substrings_vowels > total_substrings_consonants:
        print('Kevin {}'.format(total_substrings_vowels))
    else:
        print('Stuart {}'.format(total_substrings_consonants))

if __name__ == '__main__':
    s = input()
    minion_game(s)
```

BANANA
Stuart 12



Resumiendo código ...

```
suma = 0
for index, character in enumerate(string):
    if character in 'AEIOU':
        cant_subcadenas = len_string - index
        suma = suma + cant_subcadenas

# Nro total de substrings que empiezan con VOCAL
total_substrings_vowels = suma
```

```
total_substrings_vowels = sum(len_string - index for index, character in enumerate(string) if character in 'AEIOU')
```

```
# Resultado:
if total_substrings_vowels == total_substrings_consonants:
    print('Draw')
elif total_substrings_vowels > total_substrings_consonants:
    print('Kevin {}'.format(total_substrings_vowels))
else:
    print('Stuart {}'.format(total_substrings_consonants))
```

```
print(['Draw', 'Kevin {}'.format(total_substrings_vowels),
      'Stuart {}'.format(total_substrings_consonants)]
      [0 if total_substrings_vowels == total_substrings_consonants
       else 1 if total_substrings_vowels > total_substrings_consonants else 2])
```




Resolución Resumida



```
def minion_game(string):
    string = string.upper()

    len_string = len(string)
    total_substrings = (len_string)*(len_string+1)/2

    total_substrings_vowels = sum(len_string-index for index, character in enumerate(string) if character in 'AEIOU')

    total_substrings_consonants = int(total_substrings) - int(total_substrings_vowels)

    print(['Draw', 'Kevin {}'.format(total_substrings_vowels),
          'Stuart {}'.format(total_substrings_consonants)
          ][0 if total_substrings_vowels == total_substrings_consonants
            else 1 if total_substrings_vowels > total_substrings_consonants else 2])

if __name__ == '__main__':
    s = input()
    minion_game(s)
```

BANANA
Stuart 12