

Introducción a la programación

Práctica 8: Colas y Diccionarios

3/Junio/2024

Colas

Ejercicio 13

Ejercicio 13 Implementar una función

`generar_nros_al_azar(in cantidad : int, in desde : int, in hasta : int)` que genere una cola de *cantidad* de números enteros al azar en el rango $[desde, hasta]$. Pueden usar la función `random.randint(< desde >, < hasta >)` y la clase `Queue()` que es un ejemplo de una implementación básica de una Cola:

```
from queue import Queue as Cola

c = Cola()
c.put(1) # encolar
elemento = c.get() # desencolar()
c.empty() # vacia?
```

Ejercicio 16

Ejercicio 16 Bingo: un cartón de bingo contiene 12 números al azar en el rango $[0, 99]$.

1. implementar una función

`armar_secuencia_de_bingo()` \rightarrow `Cola[int]` que genere una cola con los números del 0 al 99 ordenados al azar.

2. implementar una función

`jugar_carton_de_bingo(in carton : list[int],
in bolillero : Cola[int])` \rightarrow `int` que toma un cartón de Bingo y una cola de enteros (que corresponden a las bolillas numeradas) y determina cual es la cantidad de jugadas de ese bolillero que se necesitan para ganar.

Diccionarios

Ejercicio 19

Ejercicio 19 Leer un archivo de texto y agrupar la cantidad de palabras de acuerdo a su longitud. Implementar la función `agrupar_por_longitud(in nombre_archivo : str) → dict` que devuelve un diccionario

`{longitud_en_letras : cantidad_de_palabras}`.

Ej el diccionario

```
{  
    1: 2,  
    2: 10,  
    5: 4  
}
```

indica que se encontraron 2 palabras de longitud 1, 10 palabras de longitud 2 y 4 palabras de longitud 5. Para este ejercicio vamos a considerar palabras a todas aquellas secuencias de caracteres que no tengan espacios en blanco.

Ejercicio 21

Ejercicio 21

Implementar la función

`la_palabra_mas_frecuente(in_nombre_archivo : str) → str`
que devuelve la palabra que más veces aparece en un archivo de texto. Se aconseja utilizar un diccionario de palabras para resolver el problema.