

HOJA DE PRUEBA ESCRITA

FACULTAD: FECHA: . 28/09/2021

NOMBRES Y APELLIDO: WACKER, GERARDO C..... MATRÍCULA: 000-17-3126..

ASIGNATURA: PROGRAMACIÓN I.....

CARRERA Y ORIENTACIÓN:.....

TURNO: COMISIÓN: FIRMA PROFESOR:

.....

TEMA:

Ejercicio 1:

El formato usado para las patentes consiste de 5 caracteres. Cada uno de esos caracteres es un dígito entre 0 y 9. Por ejemplo: “00123”, “99180”. La patente siguiente de una dada es la que se obtiene avanzando el último dígito, es decir, reemplazándolo por el dígito siguiente correspondiente. Por ejemplo: la patente siguiente de “00151” es “00152”. Cuando el último dígito es 9, se sustituye por el 0 y pasa a avanzar el anteúltimo dígito de la patente, teniendo en cuenta que también puede ser 9, este proceso se continúa hasta conseguir la siguiente patente. Ejemplo: la siguiente de “09199” es “09200” y la siguiente de “09999” es “10000”. (No se ingresan patentes mayores a “09999”, es decir, la última patente es “10000”).

Se pide realizar la función **siguiente(patente)** que dada una patente calcule y retorne la patente siguiente. Usar esta función para obtener la k-ésima patente siguiente. Por ejemplo: siguiente(“00151”) → “00152” y si $k = 5$, la patente 5-ésima de “00151” es “00156”

Ejercicio 2:

Se tienen tres pilas de cartas: una con cartas sólo de espada, otra con cartas sólo de oros y otra con cartas sólo de bastos. Se desea armar una cuarta pila con la siguiente combinación: oro, oro, espada, basto, basto. Tener en cuenta que las tres pilas originales pueden no tener igual cantidad de cartas.

Ejercicio 3:

Esta aplicación deberá descubrir el refrán que está escrito en una frase donde cada palabra que forma la frase está modificada de la siguiente forma:

Las palabras con una cantidad par de letras están invertidas. Por ejemplo: **sarbalap** es **palabras**

Se pide hacer la función **obtenerPalabra(palabra)** que dada una palabra obtenga la correspondiente al refrán. Por ejemplo:

obtenerPalabra(“sarbalap”) → “palabras”

obtenerPalabra(“cosmpioli”) → “olimpicos”

Usar la función para descubrir todas las palabras de un refrán. El refrán se debe poder leer de corrido, finalizado con punto y comenzando la primer letra en mayúscula. Por ejemplo: la información ingresada es: **on por mucho ragurdam amanece mas onarpmet** y el programa deberá mostrar: **No por mucho madruguar amanece mas temprano.**

Desarrollo:

1)

```
1  # Resolución del ejercicio 1 del examen parcial de Programación I
2  # @author Gerardo Wacker
3
4  # Esta función se dedica a calcular la k-ésima de una patente. Obtiene como parámetros a la patente (en formato string),
5  # y a k (en formato int). Es usada por la función "siguiente()", también solicitada.
6  def k_esima(p, k):
7      # Se solicita la patente en formato string para verificar si su longitud equivale a 5 caracteres.
8      if len(p) == 5:
9          # Si cumple con ello, se lo convierte a integer, para luego procesarlo de manera fácil.
10         patente_int = int(p)
11         # Si la suma de patente + k da un número dentro del rango solicitado, es decir, es mayor a 00000 y
12         # menor a 10000, continuar.
13         if 10000 >= (patente_int + k) >= 0:
14             # Sumar los dos parámetros, y convertirlo a string, para poder luego rellenar con los "0" que sean
15             # necesarios, y así cumplir con el formato de las patentes de 5 caracteres.
16             patente_string = str(patente_int + k)
17             # Si la k equivale a 1, se deduce que trata de obtener la siguiente. Entonces, devuelve un string
18             # redactado de la manera correspondiente.
19             if k == 1:
20                 return "La patente siguiente es " + patente_string.zfill(5) + "."
21             # Si k no es 1, entonces devolver un string que muestra el valor de k.
22             else:
23                 return "La " + str(k) + "-ésima patente es " + patente_string.zfill(5) + "."
24             # Finalmente, en caso de que no cumpla con uno de los parámetros solicitados, devolver un string aclarándolo.
25             else:
26                 return "La patente no es válida, el resultado no se encuentra dentro del rango establecido (00000 a 10000)."
27         else:
28             return "La patente no es válida, debe estar compuesta por 5 caracteres."
29
30
31 # Esta función se dedica a calcular la siguiente patente, por lo que es
32 # simplemente calcular la k-ésima, pero siendo k = 1.
33 def siguiente(p):
34     return k_esima(p, 1)
35
36
37 # Inputs correspondientes para que las dos funciones puedan correr, una detrás de otra.
38 patente = input("Ingresa la patente: ")
39 print(siguiente(patente))
40
41 patente2 = input("Ingresa una segunda patente: ")
42 input_k = input("Ahora, ingresa cuántas patentes más querés ver: ")
43 print(k_esima(patente2, int(input_k)))
44
```

(Por si acaso, se adjunta el archivo Ejercicio1.py)

2)

```
1  Pila A = Pila de cartas a ser usada
2  Pila B = Pila de cartas de palo Oro
3  Pila C = Pila de cartas de palo Espada
4  Pila D = Pila de cartas de palo Basto
5
6  comenzar ejercicio_2
7      mientras la pila B no está vacía y la pila C no está vacía y la pila D no está vacía hacer
8          tomar una carta de la pila B
9          depositar la carta en la pila A
10
11         tomar una carta de la pila B
12         depositar la carta en la pila A
13
14         tomar una carta de la pila C
15         depositar la carta en la pila A
16
17         tomar una carta de la pila D
18         depositar la carta en la pila A
19
20         tomar una carta de la pila D
21         depositar la carta en la pila A
22     finmientras
23 fin ejercicio_2
24
```

(Por si acaso, se adjunta el archivo Ejercicio2.txt)

3)

```
1  # Resolución del ejercicio 3 del examen parcial de Programación I
2  # @author Gerardo Wacker
3
4  # Función solicitada para poder obtener la palabra correspondiente. Si la misma es par, devolverla al revés. Si no,
5  # devolverla en su orden original.
6  def obtenerPalabra(palabra):
7      if len(palabra) % 2 == 0:
8          return palabra[::-1]
9      else:
10         return palabra
11
12
13 # Función principal para descubrir el contenido del refrán.
14 def descubrirRefran(refran):
15     # Declarar el resultado como string vacío, al cual luego vamos a agregarle contenido.
16     resultado = ""
17     # Dividir las palabras del refrán en una lista.
18     palabras = refran.split()
19     # Se hace un for, iterando por números. Se hace esto para poder saber si, después de cada palabra,
20     # se le agrega un espacio, o un punto final.
21     for i in range(len(palabras)):
22         # Establecemos el valor de palabra, mediante obtener el valor en la posición i. Al mismo tiempo, quitamos de
23         # la palabra los puntos, y la haremos minúscula para que tenga efecto la capitalización del final.
24         palabra = palabras[i].lower().replace(".", "")
25         # Usando la función obtenerPalabra, se agrega al string final la palabra en su formato correcto.
26         resultado += obtenerPalabra(palabra)
27         # Procesamiento previamente descrito, donde si se trata de la última palabra en el refrán, se agregará un
28         # punto en lugar de un espacio.
29         if i == (len(palabras) - 1):
30             resultado += "."
31         else:
32             resultado += " "
33     # Finalmente, hacemos mayúscula la primera letra del refrán.
34     return resultado.capitalize()
35
36
37 # Input correspondiente para obtener el refrán.
38 refran = input("Ingresa el refrán: ")
39 print(descubrirRefran(refran))
40
```

(Por si acaso, se adjunta el archivo Ejercicio3.py)