Fundamentos de Programación

Cuaderno de Trabajo 6 - Recursividad

Grado en Sistemas de la Información, turno de mañana

Grado en Ingeniería Informática, turnos de mañana y tarde

Ejercicios Resueltos

1. Realice un programa que lea desde teclado un número y calcule su factorial empleando recursividad.

SOLUCION PROPUESTA:

```
def factorial(n):
    """
    int -> long
    OBJ: funcion factorial n!
    PRE: n >= 0
    """
    if n in (0,1):
      result = 1
    else:
      result = n * factorial(n-1)
    return result

# Prueba
x = int(input("Introduzca un numero: "))
try:
  print(factorial(x))
except: print("Error, el número introducido no es válido")
```

2. Implemente una función recursiva que dada una lista averigüe si todos los elementos de la misma son iguales a uno dado.

SOLUCION PROPUESTA:

```
def todos_iguales(A, ele, actual):
    """
    lista, objeto, int -> bool
    OBJ: comprueba si todos los elementos de una lista son iguales a uno dado
    PRE: La lista debe contener al menos un elemento
    """
    if (actual == len(A)-1):
      son_iguales = (A[actual] == ele)
    else:
      son_iguales = (A[actual] == ele) and (todos_iguales(A,ele,actual+1))
    return son_iguales

# Prueba
L = [1,1,1,1,1]
print(todos_iguales(L,1,0)) # Debe mostrar True
L2 = [1,2,3,1,1]
print(todos_iguales(L2,1,0)) # Debe mostrar False
```

SOLUCION PROPUESTA (utilizando *slicing*):

```
def todos_iguales(A, ele):
    """
    lista, objeto -> bool
    OBJ: comprueba si todos los elementos de una lista son iguales a uno dado
    PRE: La lista debe contener al menos un elemento
    """
    if (len(A) == 1):
      result = (A[0] == ele)
    else:
      result = (A[0] == ele) and (todos_iguales(A[1:len(A)],ele))
    return result

# Prueba
L1 = [1,1,1,1,1]
print(todos_iguales(L1,1,0)) # Debe mostrar True
L2 = [1,2,3,1,1]
print(todos_iguales(L2,1,0)) # Debe mostrar False
```

Ejercicios propuestos

1. Programar un procedimiento recursivo que compruebe si un cierto número se encuentra o no en una lista.
2. Implementar un programa que dados dos números, calcule el producto de forma recursiva. Los números a multiplicar deben ser leídos por teclado. NOTA: no puede utilizar el operador de multiplicación así que utilice sumas.
3. Muy similar al anterior: Programar ahora un algoritmo recursivo que permita hacer una división entera mediante restas sucesivas.
4. Programar, haciendo uso de la recursividad, una función en Python que permita obtener el término de orden n de la sucesión de Fibonacci (https://es.wikipedia.org/wiki/Sucesi%C3%B3n_de_Fibonacci)
5. Programar una función que dada una palabra, retorne la misma invertida utilizando para ello recursividad.
6. Realizar una función recursiva que dado un número entero, cuente su número de dígitos.
7. Realizar un programa que lea desde teclado un número entero positivo y lo convierta a binario utilizando recursividad.
8. Calcular la suma de los números pares entre 0 y n de manera recursiva.
9. Programar un algoritmo recursivo que obtenga la suma de las edades de todos los elementos de una lista de alumnos. Un alumno está caracterizado por tres atributos (*nombre, edad, titulacion*).
10. Dado un número en base decimal y una base menor que diez, escribir un programa que cambie dicho número a la base dada utilizando para ello un procedimiento recursivo.