# **Taller 4- Algoritmos recursivos vs iterativos**

1. ¿Qué es un algoritmo recursivo?

**R/.**

Se basan en la metodología de llamar repetidamente la propia función en que están definidos, y son de gran utilidad en multitud de campos en la informática. Técnica divide y vencerás, dividir el problema en problemas más pequeños

1. Qué es un algoritmo iterativo

**R/.**

Aquellos que llegan a un resultado a través de una iteración mediante un ciclo definido o indefinido.

1. Cuáles son las principales diferencias entre un algoritmo iterativo y uno recursivo

**R/.**

* El algoritmo iterativo obtiene el resultado, mientras que el recursivo va haciendo varios procesos para llegar al mismo resultado
* Uso de más memoria del algoritmo recursivo a diferencia del iterativo

1. cómo funciona un algoritmo recursivo de un ejemplo

def factor(n):

if n==0 or n ==1:

return 1

else:

return n\*factor(n-1)

print("Fact: "+str(factor(5)))

Donde la función lo que hace es encontrar el numero más pequeño de la operación llegando así en este caso hasta el 0 o el número 1 y desde allí se empieza a multiplicar

1. cómo funciona un algoritmo iterativo de un ejemplo

n=int(input("Introduce numero: "))

x=1

for i in range(1,n+1):

x=x\*i

print("Fact: "+str(x))

En este caso siendo el mismo ejemplo va cambiando el valor de x hasta llegar a n+1, y así resuelve el problema del factorial

1. explique qué tipos de recursividad existen en qué consiste cada tipo y de un ejemplo por cada uno

**R/.**

* **Directa:** Función se llama a si misma una o varias veces
* **Indirecta:** Función es llamada de manera indirecta, es decir, por medio de otra función

**Factorial**

#Factorial iter

n=int(input("Introduce numero: "))

x=1

for i in range(1,n+1):

x=x\*i

print("Fact: "+str(x))

#Factorial recursivo

def factor(n):

if n==0 or n ==1:

return 1

else:

return n\*factor(n-1)

# **Prueba de escritorio**

**Factorial**

|  |  |  |
| --- | --- | --- |
| **Instrucciones** | **entrada** | **procesos** |
| leer entrada | 7 |  |
| x=1 |  |  |
| i in range(1,n+1) |  | i(1,8) |
| **i=1** |  | 1 |
| x |  | 1 |
| **i=2** |  | 2 |
| x |  | 2 |
| **i=3** |  | 3 |
| x |  | 6 |
| **i=4** |  | 4 |
| x |  | 24 |
| **i=5** |  | 5 |
| x |  | 120 |
| **i=6** |  | 6 |
| x |  | 720 |
| **i=7** |  | 7 |
| x |  | 5040 |
| **i=8** |  | 8 |

**Sumarreglo**

|  |  |  |
| --- | --- | --- |
| **Instrucción** | **entrada** | **proceso** |
| **leer** entrada | 5 |  |
| longitud=entrada |  | l=5 |
| i=0;i<num;i++ |  |  |
| **leer** numero | {5,7,9,10,6} |  |
| conta=0 |  |  |
| conta<5 |  |  |
| conta++ |  |  |
|  |  | total+=num[pos] |
|  |  | total=5 |
| **conta=0** |  | 5 |
| 1 |  | 12 |
| 2 |  | 21 |
| 3 |  | 31 |
| 4 |  | 37 |
| **pirnt** |  | res=37 |

# **Parte 2**

**Palindromo**

N=int(input("Ingresar numero de pruebas: "))

if 0<=N<=100:

for i in range(0,N):

frase=input("Ingrese frase a evaluar: ")

cadenaInvertida = frase[::-1]

if frase == cadenaInvertida:

print("Palindromo")

else:

print("No palindromo")

**Permutaciones**

def perms(cade):

if(len(cade)==1): return [cade]

result=[]

for i,v in enumerate(cade):

result += [v+p for p in perms(cade[:i]+cade[i+1:])]

return result

n=input("Ingrese cadena: ")

print('\n'.join(perms(n)))

**Canción sin vocales**

def convertir(lista):

cad=""

for ele in lista:

cad += ele

return(cad)

voc='aeiouAEIOU'

cons='bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'

f = open ('C:/Users/brian/Documents/Análisis/Canc.txt','r')

mss = f.read()

lista=len([m for m in mss if m in voc])

print(mss)

print("Cantidad de vocales: "+str(lista))

sv= open ('C:/Users/brian/Documents/Análisis/Canc\_sinvoc.txt','w')

lis=[m for m in mss if m in cons]

print(convertir(lis))

men=sv.write(convertir(lis))

sv.close()

f.close()

**Poema**

voc='aeiouAEIOU'

espaci=' '

cons='bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'

let='aeiouAEIOUbcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'

f = open ('C:/Users/brian/Documents/Análisis/Poet.txt','r')

mss = f.read()

print(mss)

def contar(var):

lista=len([m for m in mss if m in var])

return lista

p=contar(voc)

print("Cantidad de vocales: "+str(p))

es=contar(espaci)

print("Cantidad de espacio: "+str(es))

c=contar(cons)

print("Cantidad de consonantes: "+str(c))

l=contar(let)

print("Cantidad de letras: "+str(l))

fr=mss.split()

pal=[]

for i in fr:

pal.append(fr.count(i))

print("Frase\n" + str(list(zip(fr, pal))))

f.close()