

Tarea

PIAD-416 Algoritmia para la Inteligencia Artificial

Ingeniería de Software con Inteligencia Artificial Semestre IV



TAREA N°2

Utiliza recursividad en la programación

1. Implemente, tanto de forma recursiva como de forma iterativa, una función que nos diga si una cadena de caracteres es simétrica (un palíndromo). Por ejemplo, "DABALEARROZALAZORRAELABAD" es un palíndromo.

CÓDIGO:

```
def es_palindromo_recursivo(cadena):
    #caso base
    if len(cadena) <=1:
        return True

    #caso recursivo

    if cadena[0] == cadena[-1]:
        return es_palindromo_recursivo(cadena[1:-1])
    else:
        return False

cadena = "DABALEARROZALAZORRAELABAD"
es_palindromo = es_palindromo_recursivo(cadena)
if es_palindromo:
    print("la cadena es un palíndromo")
else:
    print("la cadena no es un palíndromo")
```

EXPLICACIÓN:

Algoritmia para la Inteligencia Artificial

.EXPLICACIÓN:

```
def es_palindromo_recursivo(cadena):
```

/Se está definiendo la función "es_palindromo _recursivo " con el argumento " cadena"

```
if len(cadena) <=1:
```

```
    return True
```

/se usa "if" y "len" para decir que si la longitud de la cadena es menor o igual a uno ya que si solo tiene un dígito o 0 dígitos, automáticamente sería un palíndromo y arrojaría un valor True.

```
    if cadena[0] == cadena[-1]:
```

/"cadena[0]" hace referencia al primer dígito de la cadena, "cadena[-1]" hace referencia al ultimo dígito de la cadena; esto quiere decir que esta verificando si el primer y el ultimo dígito son iguales.

```
        return es_palindromo_recursivo(cadena[1:-1])
```

```
    else:
```

```
        return False
```

/si el primer y el ultimo digito son iguales retorna "es_palindromo_recursivo(cadena[1:-1])", osea comprueba que es una afirmación verdadera, d elo contrario tendremos un valor False.

```
cadena = "DABALEARROZALAZORRAELABAD"
```

```
/esta es la cadena/
```

```
es_palindromo = es_palindromo_recursivo(cadena)
```

/"es_palindromo" es una variable que toma el valor de "es_palindromo_recursivo(cadena)"/

```
if es_palindromo:
```

```
    print("la cadena es un palíndromo")
```

```
else:
```

```
    print("la cadena no es un palíndromo")
```

/aquí nos dice que si es verdadero , que imprima "la cadena es un palíndromo" sino que imprima "la cadena no es un palíndromo".

2. Escribir una función recursiva que devuelva la suma de los primeros N enteros.

CÓDIGO:

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)

resultado = fibonacci(3)
print(resultado)
```

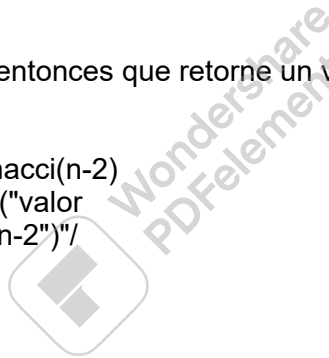
EXPLICACIÓN:

```
def fibonacci(n):
    /se define la función fibonacci con un argumento "n"/

    if n== 0
        return = 0
    /si n es igual a 0, entonces que retorne un valor o/

    elif n == 1
        return 1
    /en caso que n sea igual a 1, entonces que retorne un valor de 1/

    else:
        return fibonacci(n-1) + fibonacci(n-2)
        /sino que retorne "fibonacci(valor
        de n-1") + fibonacci("valor de n-2")/
```



3. Programe un método recursivo que transforme un número expresado en notación binaria a un número entero.

CÓDIGO:

```
def binario_a_entero(binario):
    if len(binario) == 0:
        return 0
    else:
        ultimo_digito = int(binario[-1])
        potencia = len(binario) - 1
        return ultimo_digito * (2 ** potencia) + binario_a_entero(binario[:-1])

resultado = binario_a_entero
numero_binario = "101010"
numero_entero = binario_a_entero(numero_binario)
print(numero_entero)
```

EXPLICACIÓN:

```
def binario_a_entero(binario)
/define la función "binario_a_entero" con un argumento "binario"/

if len(binario) == 0
    return 0
/is la longitud de la longitud de los dígitos es igual a 0, que retorne un valor 0 /

else:
    ultimo_digito = int(binario[-1])
    /crea la variable "ultimo_digito" y toma el último dígito de la cadena, y lo
    transforma a un dato entero/

    potencia = len(binario) - 1
    /"crea la variable potencia", y calcula a que puesto o a que longitud se
    encuentra/

    return ultimo_digito * (2 ** potencia) + binario_a_entero(binario[:-1])/
/retorna el valor del último dígito multiplicado por 2 por la potencia, elevado al
potencia de su posición en la cadena, todo esto lo suma al valor de la función "
binario_a_entero", pero con el último dígito eliminado/
```