

ACTIVIDADES

Algoritmo: Cálculo del factorial de un número

Definición: ¡La factorial de un número entero positivo n , denotado por $n!$, es el producto de todos los números enteros positivos menores o iguales a n .

Ejemplo: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Código Python:

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
  
n = int(input("Ingrese un número entero positivo: "))  
print("El factorial de", n, "es:", factorial(n))
```

Explicación:

1. La función `factorial(n)` calcula el factorial de un número entero positivo n .
2. Si n es 0, la función devuelve 1, ya que el factorial de 0 es 1.
3. Si n es mayor que 0, la función devuelve el producto de n y el factorial de $n-1$.
4. El programa pide al usuario que ingrese un número entero positivo.
5. El programa llama a la función `factorial(n)` con el número ingresado y imprime el resultado

Ejemplo de ejecución:

Ingrese un número entero positivo: 5 El factorial de 5 es: 120

ACTIVIDADES

¿Cuántos términos de la secuencia de Fibonacci deseas? 10

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

Este es un algoritmo matemático sencillo, pero también muy representativo en la programación y puede extenderse a problemas más complejos.

Función para generar la secuencia de Fibonacci

```
def fibonacci(n):
```

```
    fib_sequence = []
```

```
    a, b = 0, 1
```

```
    for _ in range(n):
```

```
        fib_sequence.append(a)
```

```
        a, b = b, a + b
```

```
    return fib_sequence
```

Solicitar al usuario la cantidad de términos

```
n = int(input("¿Cuántos términos de la secuencia de Fibonacci deseas? "))
```

Llamar a la función e imprimir la secuencia

```
print(fibonacci(n))
```

ACTIVIDADES

Crear usuario y contraseña github

Actividades

1. Código Numero

INICIO

SUMA \leftarrow 0

PAR \leftarrow 2

MIENTRAS PAR \leq 50 HACER

SUMA \leftarrow SUMA + PAR

PAR \leftarrow PAR + 2

FIN MIENTRAS

ESCRIBIR "La suma de los números pares entre 1 y 50 es: " + SUMA

FIN

ACTIVIDADES:

```
# Elegir una palabra aleatoria palabra_oculta = random.choice(palabras)
```

```
# Inicializar variables letras_adivinadas = ['_'] * len(palabra_oculta) intentos = 6
```

```
# Número de intentos permitidos
```

```
# Mostrar la palabra oculta con guiones print("¡Bienvenido al juego de adivinar la palabra!")
print("La palabra tiene", len(palabra_oculta), "letras.") print(" ".join(letras_adivinadas))
```

```
# Comenzar el juego while intentos > 0: letra = input("\nAdivina una letra: ").lower() if len(letra) !=
1 or not letra.isalpha(): print("Por favor, ingresa solo una letra.") continue
```

```
# Verificar si la letra está en la palabra oculta if letra in palabra_oculta: print(f"¡Bien hecho! La
letra '{letra}' está en la palabra.")
```

```
# Actualizar las letras adivinadas for i in range(len(palabra_oculta)): if palabra_oculta[i] == letra:
letras_adivinadas[i] = letra else: intentos -= 1 print(f"Lo siento, la letra '{letra}' no está en la
palabra.") print(f"Te quedan {intentos} intentos.")
```

```
# Mostrar las letras adivinadas print(" ".join(letras_adivinadas)) # Verificar si el jugador ha
adivinado la palabra completa if "_" not in letras_adivinadas: print("¡Felicidades! Has adivinado la
palabra:", palabra_oculta) break
```

```
# Si se acabaron los intentos, el jugador pierde if intentos == 0: print("¡Se acabaron los intentos!
La palabra era:", palabra_oculta)
```