

# Instrucciones para el Laboratorio

Para cada ejercicio del laboratorio, se requiere seguir estos pasos:

1. **Desarrollo del Pseudocódigo:** Antes de comenzar a programar, elabora un pseudocódigo para cada solución. El pseudocódigo debe detallar la lógica y los pasos necesarios para resolver el problema de manera clara y estructurada. Esto te ayudará a comprender mejor la solución antes de codificarla.
2. **Implementación del Programa:** Una vez que tengas el pseudocódigo listo y revisado, procede a programar la solución en Python. Asegúrate de que tu código siga la lógica que has definido en el pseudocódigo y cumpla con los requisitos especificados en cada ejercicio.

Recuerda que una buena planificación a través del pseudocódigo facilitará el proceso de programación y te permitirá identificar posibles errores antes de implementar el código.

## Ejercicio: Transformación de Palabras

Dada la lista de palabras:

```
words = ['play', 'filling', 'bar', 'theatre', 'easygoing', 'date', 'lead', 'that', 'story', 'island']
```

1. **Imprimir Letras Mayúsculas:** Escribe un programa que imprima todas las letras de cada palabra en la lista en mayúsculas, separando cada letra por una coma.

```
# tu pseudocodigo aquí

# Para cada palabra en la lista 'words' hacer lo siguiente:
#   Inicializar una cadena vacía llamada 'resultado'
#   Para cada letra en la palabra hacer lo siguiente:
#       Convertir la letra a mayúsculas
#       Añadir la letra mayúscula a 'resultado' con una coma después
#   de cada letra
#   Si 'resultado' no está vacío, eliminar la última coma de
#   'resultado'
#   Imprimir 'resultado'

# tu codigo con for aquí

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    resultado = "" # Inicializar una cadena vacía
    # Para cada letra en la palabra hacer lo siguiente:
    for letter in word:
        resultado += letter.upper() + "," # Convertir a mayúsculas y
# añadir coma
```

```

if resultado: # Si 'resultado' no está vacío
    resultado = resultado[:-1] # Eliminar la última coma

print(resultado) # Imprimir 'resultado'

P,L,A,Y
F,I,L,L,I,N,G
B,A,R
T,H,E,A,T,R,E
E,A,S,Y,G,O,I,N,G
D,A,T,E
L,E,A,D
T,H,A,T
S,T,O,R,Y
I,S,L,A,N,D

# tu codigo con while aquí

# Inicializar un índice para recorrer la lista
index_palabra = 0 # Inicializar el índice para la lista de palabras

# Mientras el índice de palabra sea menor que la longitud de la lista
# 'words':
while index_palabra < len(words):
    word = words[index_palabra] # Obtener la palabra actual
    resultado = "" # Inicializar una cadena vacía
    index_letra = 0 # Inicializar el índice para la palabra actual

    # Mientras el índice de la letra sea menor que la longitud de la
    # palabra actual:
    while index_letra < len(word):
        resultado += word[index_letra].upper() + "," # Convertir a
        mayúsculas y añadir coma
        index_letra += 1 # Incrementar el índice de la letra

    if resultado: # Si 'resultado' no está vacío
        resultado = resultado[:-1] # Eliminar la última coma

    print(resultado) # Imprimir 'resultado'

    index_palabra += 1 # Incrementar el índice de la palabra

P,L,A,Y
F,I,L,L,I,N,G
B,A,R
T,H,E,A,T,R,E
E,A,S,Y,G,O,I,N,G
D,A,T,E
L,E,A,D
T,H,A,T

```

S,T,O,R,Y  
I,S,L,A,N,D

1. **Crear una Nueva Lista:** Genera una nueva lista que contenga solo las palabras que tengan 5 o más letras. Luego, imprime esta nueva lista.

```
# tu pseudocodigo aquí

# Definir la lista de palabras
# Inicializar una lista vacía llamada 'nueva_lista'
# Para cada palabra en la lista 'words' hacer lo siguiente:
#     Si la longitud de la palabra es mayor o igual a 5:
#         Añadir la palabra a 'nueva_lista'
# Imprimir 'nueva_lista'

# tu codigo con for aquí

# Inicializar una lista vacía llamada 'nueva_lista'
nueva_lista = []

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    # Si la longitud de la palabra es mayor o igual a 5:
    if len(word) >= 5:
        # Añadir la palabra a 'nueva_lista'
        nueva_lista.append(word)

# Imprimir 'nueva_lista'
print(nueva_lista)

['filling', 'theatre', 'easygoing', 'story', 'island']

# tu codigo con while aquí

# Inicializar una lista vacía llamada 'nueva_lista'
nueva_lista = []

# Inicializar el índice para el bucle while
index = 0

# Mientras el índice sea menor que la longitud de la lista 'words':
while index < len(words):
    word = words[index] # Obtener la palabra actual
    # Si la longitud de la palabra es mayor o igual a 5:
    if len(word) >= 5:
        # Añadir la palabra a 'nueva_lista'
        nueva_lista.append(word)
    index += 1 # Incrementar el índice

# Imprimir 'nueva_lista' nuevamente para mostrar palabras agregadas
print(nueva_lista)
```

```
['filling', 'theatre', 'easygoing', 'story', 'island']
```

1. **Encontrar Palabras que Comienzan con 't':** Imprime la primera palabra en la lista que comience con la letra 't'. Si no hay ninguna, imprime un mensaje indicando que no se encontró ninguna palabra que empiece con 't'.

```
# tu pseudocodigo aquí

# Definir la lista de palabras
# Inicializar una variable para almacenar la palabra encontrada
# Para cada palabra en la lista 'words' hacer lo siguiente:
#     Si la palabra comienza con 't':
#         Asignar la palabra a 'palabra_encontrada'
#         Salir del bucle

# Si 'palabra_encontrada' es diferente de None:
#     Imprimir 'palabra_encontrada'
# Sino:
#     Imprimir un mensaje indicando que no se encontró ninguna palabra
#     que empiece con 't'

# tu codigo con for aquí

# Inicializar una variable para almacenar la palabra encontrada
palabra_encontrada = None

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    # Si la palabra comienza con 't':
    if word.startswith('t'):
        # Asignar la palabra a 'palabra_encontrada'
        palabra_encontrada = word
        break # Salir del bucle una vez que se encuentra la primera
palabra

# Si 'palabra_encontrada' es diferente de None:
if palabra_encontrada is not None:
    # Imprimir 'palabra_encontrada'
    print(palabra_encontrada)
else:
    # Imprimir un mensaje indicando que no se encontró ninguna palabra
    # que empiece con 't'
    print("No se encontró ninguna palabra que empiece con 't'.")

theatre

# tu codigo con while aquí

# Inicializar una variable para almacenar la palabra encontrada
palabra_encontrada = None
```

```

# Inicializar el índice para el bucle while
index = 0

# Mientras el índice sea menor que la longitud de la lista 'words':
while index < len(words):
    word = words[index] # Obtener la palabra actual
    # Si la palabra comienza con 't':
    if word.startswith('t'):
        # Asignar la palabra a 'palabra_encontrada'
        palabra_encontrada = word
        break # Salir del bucle una vez que se encuentra la primera
palabra
        index += 1 # Incrementar el índice

# Si 'palabra_encontrada' es diferente de None:
if palabra_encontrada is not None:
    # Imprimir 'palabra_encontrada'
    print(palabra_encontrada)
else:
    # Imprimir un mensaje indicando que no se encontró ninguna palabra
    que empiece con 't'
    print("No se encontró ninguna palabra que empiece con 't'.")

theatre

```

1. **Contar las Vocales:** Escribe un programa que cuente e imprima el número total de vocales en todas las palabras de la lista.

```

# tu pseudocódigo aquí

# Definir la lista de palabras
# Inicializar un contador de vocales en 0
# Para cada palabra en la lista 'words' hacer lo siguiente:
#     Para cada letra en la palabra hacer lo siguiente:
#         Si la letra es una vocal (a, e, i, o, u):
#             Incrementar el contador de vocales en 1
# Imprimir el número total de vocales

# tu código con for aquí

# Inicializar un contador de vocales en 0
contador_vocales = 0

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    # Para cada letra en la palabra hacer lo siguiente:
    for letter in word:
        # Si la letra es una vocal (a, e, i, o, u):
        if letter.lower() in 'aeiou':
            # Incrementar el contador de vocales en 1

```

```

        contador_vocales += 1

# Imprimir el número total de vocales
print("Número total de vocales:", contador_vocales)

Número total de vocales: 19

# tu codigo con while aquí

# Inicializar un contador de vocales en 0
contador_vocales = 0

# Inicializar el índice para el bucle while
index = 0

# Mientras el índice sea menor que la longitud de la lista 'words':
while index < len(words):
    word = words[index] # Obtener la palabra actual
    # Inicializar el índice de letras para cada palabra
    letter_index = 0
    # Mientras el índice de letras sea menor que la longitud de la
    palabra:
        while letter_index < len(word):
            # Si la letra es una vocal (a, e, i, o, u):
            if word[letter_index].lower() in 'aeiou':
                # Incrementar el contador de vocales en 1
                contador_vocales += 1
            letter_index += 1 # Incrementar el índice de letras
        index += 1 # Incrementar el índice de palabras

# Imprimir el número total de vocales
print("Número total de vocales:", contador_vocales)

Número total de vocales: 19

```

1. **Palabras que Contienen la Letra 'a':** Crea una nueva lista que contenga solo las palabras que tengan la letra 'a'. Imprime esta nueva lista.

```

# tu pseudocodigo aquí

# Definir la lista de palabras
# Inicializar una nueva lista vacía llamada 'nueva_lista'
# Para cada palabra en la lista 'words' hacer lo siguiente:
#     Si la letra 'a' está en la palabra:
#         Añadir la palabra a 'nueva_lista'
# Imprimir 'nueva_lista'

# tu codigo con for aquí

# Inicializar una nueva lista vacía llamada 'nueva_lista'
nueva_lista = []

```

```

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    # Si la letra 'a' está en la palabra:
    if 'a' in word:
        # Añadir la palabra a 'nueva_lista'
        nueva_lista.append(word)

# Imprimir 'nueva_lista'
print("Palabras que contienen la letra 'a':", nueva_lista)

Palabras que contienen la letra 'a': ['play', 'bar', 'theatre',
'easygoing', 'date', 'lead', 'that', 'island']

# tu código con while aquí

# Inicializar una nueva lista vacía llamada 'nueva_lista'
nueva_lista = []

# Inicializar el índice para el bucle while
index = 0

# Mientras el índice sea menor que la longitud de la lista 'words':
while index < len(words):
    word = words[index] # Obtener la palabra actual
    # Si la letra 'a' está en la palabra:
    if 'a' in word:
        # Añadir la palabra a 'nueva_lista'
        nueva_lista.append(word)
    index += 1 # Incrementar el índice

# Imprimir 'nueva_lista'
print("Palabras que contienen la letra 'a':", nueva_lista)

Palabras que contienen la letra 'a': ['play', 'bar', 'theatre',
'easygoing', 'date', 'lead', 'that', 'island']

```

1. **Palabras en Orden Inverso:** Imprime la lista de palabras en orden inverso, de la última a la primera.

```

# tu pseudocódigo aquí

# Definir la lista de palabras
# Inicializar una nueva lista vacía llamada 'nueva_lista'
# Para cada palabra en la lista 'words' hacer lo siguiente:
#     Añadir la palabra al inicio de 'nueva_lista'
# Imprimir 'nueva_lista'

# tu código con for aquí

# Inicializar una nueva lista vacía llamada 'nueva_lista'

```

```
nueva_lista = []

# Para cada palabra en la lista 'words' hacer lo siguiente:
for word in words:
    # Añadir la palabra al inicio de 'nueva_lista'
    nueva_lista.insert(0, word)

# Imprimir 'nueva_lista'
print("Palabras en orden inverso:", nueva_lista)

Palabras en orden inverso: ['island', 'story', 'that', 'lead', 'date',
'easygoing', 'theatre', 'bar', 'filling', 'play']

# tu codigo con while aquí

# Inicializar una nueva lista vacía llamada 'nueva_lista'
nueva_lista = []

# Inicializar el índice para el bucle while
index = 0

# Mientras el índice sea menor que la longitud de la lista 'words':
while index < len(words):
    # Añadir la palabra al inicio de 'nueva_lista'
    nueva_lista.insert(0, words[index])
    index += 1 # Incrementar el índice

# Imprimir 'nueva_lista'
print("Palabras en orden inverso:", nueva_lista)

Palabras en orden inverso: ['island', 'story', 'that', 'lead', 'date',
'easygoing', 'theatre', 'bar', 'filling', 'play']
```

## Numeros Primos

1. **Lista de Números Primos:** Crea una lista que contenga los números primos entre 1 y 20. Imprime la lista.

```
# tu pseudocodigo aquí

# Inicializar una nueva lista vacía llamada 'numeros_primos'
# Para cada número desde 2 hasta 20 hacer lo siguiente:
#     Inicializar una variable llamada 'es_primo' como Verdadero
#     Para cada número desde 2 hasta la raíz cuadrada del número
actual:
#         Si el número actual es divisible por este número:
#             Cambiar 'es_primo' a Falso
#     Si 'es_primo' es Verdadero:
#         Añadir el número actual a 'numeros_primos'
# Imprimir 'numeros_primos'
```



```

# tu codigo con for aquí

# Inicializar una nueva lista vacía llamada 'numeros_primos'
numeros_primos = []

# Para cada número desde 2 hasta 20 hacer lo siguiente:
for num in range(2, 21):
    es_primo = True # Inicializar como Verdadero
    # Para cada número desde 2 hasta la raíz cuadrada del número
    actual:
    for i in range(2, int(num**0.5) + 1):
        # Si el número actual es divisible por este número:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Añadir el número actual a 'numeros_primos'
        numeros_primos.append(num)

# Imprimir 'numeros_primos'
print("Números primos entre 1 y 20:", numeros_primos)

```

Números primos entre 1 y 20: [2, 3, 5, 7, 11, 13, 17, 19]

```

# tu codigo con while aquí

# Inicializar una nueva lista vacía llamada 'numeros_primos'
numeros_primos = []

# Inicializar el número actual en 2
num = 2

# Mientras el número actual sea menor o igual a 20:
while num <= 20:
    es_primo = True # Inicializar como Verdadero
    # Inicializar el divisor en 2
    i = 2
    # Mientras el divisor sea menor o igual a la raíz cuadrada del
    número actual:
    while i <= int(num**0.5):
        # Si el número actual es divisible por este divisor:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero:
    if es_primo:

```

```

    # Añadir el número actual a 'numeros_primos'
    numeros_primos.append(num)

    num += 1 # Incrementar el número actual

# Imprimir 'numeros_primos'
print("Números primos entre 1 y 20:", numeros_primos)

```

Números primos entre 1 y 20: [2, 3, 5, 7, 11, 13, 17, 19]

1. **Contar Números Primos:** Escribe un programa que cuente cuántos números primos hay entre 1 y 30. Imprime el total.

```

# tu pseudocodigo aquí

# Inicializar un contador llamado 'contador_primos' en 0
# Para cada número desde 2 hasta 30 hacer lo siguiente:
#     Inicializar una variable llamada 'es_primo' como Verdadero
#     Para cada número desde 2 hasta la raíz cuadrada del número
actual:
#         Si el número actual es divisible por este número:
#             Cambiar 'es_primo' a Falso
#     Si 'es_primo' es Verdadero:
#         Incrementar 'contador_primos' en 1
# Imprimir el valor de 'contador_primos'

# tu codigo con for aquí

# Inicializar un contador llamado 'contador_primos' en 0
contador_primos = 0

# Para cada número desde 2 hasta 30 hacer lo siguiente:
for num in range(2, 31):
    es_primo = True # Inicializar como Verdadero
    # Para cada número desde 2 hasta la raíz cuadrada del número
actual:
    for i in range(2, int(num**0.5) + 1):
        # Si el número actual es divisible por este número:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Incrementar 'contador_primos' en 1
        contador_primos += 1

# Imprimir el valor de 'contador_primos'
print("Total de números primos entre 1 y 30:", contador_primos)

```

Total de números primos entre 1 y 30: 10

```

# tu codigo con while aquí

# Inicializar un contador llamado 'contador_primos' en 0
contador_primos = 0

# Inicializar el número actual en 2
num = 2

# Mientras el número actual sea menor o igual a 30:
while num <= 30:
    es_primo = True # Inicializar como Verdadero
    # Inicializar el divisor en 2
    i = 2
    # Mientras el divisor sea menor o igual a la raíz cuadrada del
    número actual:
    while i <= int(num**0.5):
        # Si el número actual es divisible por este divisor:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Incrementar 'contador_primos' en 1
        contador_primos += 1

    num += 1 # Incrementar el número actual

# Imprimir el valor de 'contador_primos'
print("Total de números primos entre 1 y 30:", contador_primos)

Total de números primos entre 1 y 30: 10

```

1. **Suma de Números Primos:** Crea un programa que calcule la suma de los números primos menores de 50 y la imprima.

```

# tu pseudocodigo aquí

# Inicializar una variable llamada 'suma_primos' en 0
# Para cada número desde 2 hasta 49 hacer lo siguiente:
#     Inicializar una variable llamada 'es_primo' como Verdadero
#     Para cada número desde 2 hasta la raíz cuadrada del número
    actual:
#         Si el número actual es divisible por este número:
#             Cambiar 'es_primo' a Falso
#     Si 'es_primo' es Verdadero:
#         Sumar el número actual a 'suma_primos'
# Imprimir el valor de 'suma_primos'

```

```

# tu codigo con for aquí

# Inicializar una variable llamada 'suma_primos' en 0
suma_primos = 0

# Para cada número desde 2 hasta 49 hacer lo siguiente:
for num in range(2, 50):
    es_primo = True # Inicializar como Verdadero
    # Para cada número desde 2 hasta la raíz cuadrada del número
    actual:
    for i in range(2, int(num**0.5) + 1):
        # Si el número actual es divisible por este número:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Sumar el número actual a 'suma_primos'
        suma_primos += num

# Imprimir el valor de 'suma_primos'
print("La suma de los números primos menores de 50 es:", suma_primos)

```

La suma de los números primos menores de 50 es: 328

```

# tu codigo con while aquí

# Inicializar una variable llamada 'suma_primos' en 0
suma_primos = 0

# Inicializar el número actual en 2
num = 2

# Mientras el número actual sea menor de 50:
while num < 50:
    es_primo = True # Inicializar como Verdadero
    # Inicializar el divisor en 2
    i = 2
    # Mientras el divisor sea menor o igual a la raíz cuadrada del
    número actual:
    while i <= int(num**0.5):
        # Si el número actual es divisible por este divisor:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Salir del bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero:
    if es_primo:

```

```

        # Sumar el número actual a 'suma_primos'
        suma_primos += num

    num += 1 # Incrementar el número actual

# Imprimir el valor de 'suma_primos'
print("La suma de los números primos menores de 50 es:", suma_primos)

```

La suma de los números primos menores de 50 es: 328

1. **Verificar si un Número es Primo:** Escribe un programa que pida al usuario un número y que determine si es primo. Imprime un mensaje indicando si el número es primo o no.

```

# tu pseudocodigo aquí

# Pedir al usuario que ingrese un número y almacenarlo en la variable
# 'numero'
# Si 'numero' es menor que 2, imprimir que no es primo y terminar
# Inicializar una variable llamada 'es_primo' como Verdadero
# Para cada número desde 2 hasta la raíz cuadrada de 'numero':
#     Si 'numero' es divisible por este número:
#         Cambiar 'es_primo' a Falso
#         Romper el bucle
# Si 'es_primo' es Verdadero, imprimir que el número es primo
# Si no, imprimir que el número no es primo

# tu codigo con for aquí

# Pedir al usuario que ingrese un número y almacenarlo en la variable
# 'numero'
numero = int(input("Ingresa un número para verificar si es primo: "))

# Si 'numero' es menor que 2, imprimir que no es primo y terminar
if numero < 2:
    print(f"{numero} no es un número primo.")
else:
    es_primo = True # Inicializar como Verdadero

    # Para cada número desde 2 hasta la raíz cuadrada de 'numero':
    for i in range(2, int(numero**0.5) + 1):
        # Si 'numero' es divisible por este número:
        if numero % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle

    # Si 'es_primo' es Verdadero, imprimir que el número es primo
    if es_primo:
        print(f"{numero} es un número primo.")
    else:
        print(f"{numero} no es un número primo.")

```

```

Ingresa un número para verificar si es primo: 113
113 es un número primo.

# tu codigo con while aquí

# Pedir al usuario que ingrese un número y almacenarlo en la variable
'numero'
numero = int(input("Ingresa un número para verificar si es primo: "))

# Si 'numero' es menor que 2, imprimir que no es primo y terminar
if numero < 2:
    print(f"{numero} no es un número primo.")
else:
    es_primo = True # Inicializar como Verdadero
    i = 2 # Inicializar el divisor

    # Mientras el divisor sea menor o igual a la raíz cuadrada de
    'numero':
    while i <= int(numero**0.5):
        # Si 'numero' es divisible por este divisor:
        if numero % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero, imprimir que el número es primo
    if es_primo:
        print(f"{numero} es un número primo.")
    else:
        print(f"{numero} no es un número primo.")

Ingresa un número para verificar si es primo: 113
113 es un número primo.

```

1. **Primos Menores a un Número:** Crea un programa que pida al usuario un número y que imprima todos los números primos menores que ese número.

```

# tu pseudocodigo aquí

# Pedir al usuario que ingrese un número y almacenarlo en la variable
'numero'
# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
# Para cada número desde 2 hasta 'numero' - 1 hacer lo siguiente:
#     Inicializar una variable llamada 'es_primo' como Verdadero
#     Para cada número desde 2 hasta la raíz cuadrada del número
actual:
#         Si el número actual es divisible por este número:
#             Cambiar 'es_primo' a Falso

```

```

#           Romper el bucle
#   Si 'es_primo' es Verdadero:
#           Añadir el número actual a la lista 'primos'
# Imprimir la lista de números primos

# tu codigo con for aquí

# Pedir al usuario que ingrese un número y almacenarlo en la variable
'numero'
numero = int(input("Ingresa un número para encontrar todos los primos
menores que él: "))

# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
primos = []

# Para cada número desde 2 hasta 'numero' - 1 hacer lo siguiente:
for num in range(2, numero):
    es_primo = True # Inicializar como Verdadero
    # Para cada número desde 2 hasta la raíz cuadrada del número
    actual:
    for i in range(2, int(num**0.5) + 1):
        # Si el número actual es divisible por este número:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle
    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Añadir el número actual a la lista 'primos'
        primos.append(num)

```

```

# Imprimir la lista de números primos
print(f"Números primos menores que {numero}: {primos}")

```

Ingresa un número para encontrar todos los primos menores que él: 25

Números primos menores que 25: [2, 3, 5, 7, 11, 13, 17, 19, 23]

```

# tu codigo con while aquí

```

```

# Pedir al usuario que ingrese un número y almacenarlo en la variable
'numero'
numero = int(input("Ingresa un número para encontrar todos los primos
menores que él: "))

# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
primos = []

# Inicializar el número actual en 2
num = 2

```

```

# Mientras el número actual sea menor que 'numero':
while num < numero:
    es_primo = True # Inicializar como Verdadero
    i = 2 # Inicializar el divisor
    # Mientras el divisor sea menor o igual a la raíz cuadrada del
    número actual:
    while i <= int(num**0.5):
        # Si el número actual es divisible por este divisor:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Añadir el número actual a la lista 'primos'
        primos.append(num)

    num += 1 # Incrementar el número actual

# Imprimir la lista de números primos
print(f"Números primos menores que {numero}: {primos}")

```

Ingresa un número para encontrar todos los primos menores que él: 25

Números primos menores que 25: [2, 3, 5, 7, 11, 13, 17, 19, 23]

1. **Lista de Primos en un Rango:** Crea un programa que genere e imprima los números primos entre 10 y 50.

```

# tu pseudocodigo aquí

# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
# Para cada número desde 10 hasta 50 hacer lo siguiente:
#     Inicializar una variable llamada 'es_primo' como Verdadero
#     Para cada número desde 2 hasta la raíz cuadrada del número
actual:
#         Si el número actual es divisible por este número:
#             Cambiar 'es_primo' a Falso
#             Romper el bucle
#     Si 'es_primo' es Verdadero:
#         Añadir el número actual a la lista 'primos'
# Imprimir la lista de números primos

# tu codigo con for aquí

# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
primos = []

```



```

# Para cada número desde 10 hasta 50 hacer lo siguiente:
for num in range(10, 51):
    es_primo = True # Inicializar como Verdadero
    # Para cada número desde 2 hasta la raíz cuadrada del número
    actual:
    for i in range(2, int(num**0.5) + 1):
        # Si el número actual es divisible por este número:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle
    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Añadir el número actual a la lista 'primos'
        primos.append(num)

```

```

# Imprimir la lista de números primos
print(f"Números primos entre 10 y 50: {primos}")

```

Números primos entre 10 y 50: [11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

# tu codigo con while aquí

```

# Inicializar una lista vacía llamada 'primos' para almacenar los
números primos
primos = []

```

```

# Inicializar el número actual en 10
num = 10

```

```

# Mientras el número actual sea menor o igual a 50:
while num <= 50:
    es_primo = True # Inicializar como Verdadero
    i = 2 # Inicializar el divisor
    # Mientras el divisor sea menor o igual a la raíz cuadrada del
    número actual:
    while i <= int(num**0.5):
        # Si el número actual es divisible por este divisor:
        if num % i == 0:
            es_primo = False # Cambiar a Falso
            break # Romper el bucle
        i += 1 # Incrementar el divisor

    # Si 'es_primo' es Verdadero:
    if es_primo:
        # Añadir el número actual a la lista 'primos'
        primos.append(num)

    num += 1 # Incrementar el número actual

```

```
# Imprimir la lista de números primos
print(f"Números primos entre 10 y 50: {primos}")
```

```
Números primos entre 10 y 50: [11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

## People

Dada la siguiente lista de personas:

```
people = [
    {
        "name": "Juan",
        "age": 34,
        "n_kids": 2,
        "city": "Madrid"
    },
    {
        "name": "Pepe",
        "age": 27,
        "n_kids": 0,
        "city": "Barcelona"
    },
    {
        "name": "Sonia",
        "age": 41,
        "n_kids": 1,
        "city": "Valencia"
    },
    {
        "name": "Lucía",
        "age": 22,
        "n_kids": 2,
        "city": "Sevilla"
    },
    {
        "name": "Leo",
        "age": 55,
        "n_kids": 5,
        "city": "Bilbao"
    },
    {
        "name": "Marta",
        "age": 30,
        "n_kids": 3,
        "city": "Zaragoza"
    },
    {

```

```

        "name": "Carlos",
        "age": 45,
        "n_kids": 0,
        "city": "Granada"
    }
]

```

1. **Contar Personas:** Escribe un programa que cuente cuántas personas hay en la lista `people`. Imprime el total.

```

# tu pseudocodigo aquí

# Definir la lista de personas llamada 'people'
# Inicializar una variable llamada 'total_personas' en 0
# Para cada persona en la lista 'people' hacer lo siguiente:
#     Incrementar 'total_personas' en 1
# Imprimir el valor de 'total_personas'

# tu codigo con for aquí

# Inicializar una variable llamada 'total_personas' en 0
total_personas = 0

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    total_personas += 1 + person["n_kids"] # Incrementar
    'total_personas' en 1

# Imprimir el valor de 'total_personas'
print(f"Total de personas: {total_personas}")

Total de personas: 20

people[1].items()
dict_items([('name', 'Pepe'), ('age', 27), ('n_kids', 0), ('city',
'Barcelona')])

# tu codigo con while aquí

# Inicializar una variable llamada 'total_personas' en 0
total_personas = 0
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):
    total_personas += 1 + people[index]["n_kids"] # Incrementar
    'total_personas' en 1
    index += 1 # Incrementar el índice

```

```
# Imprimir el valor de 'total_personas'
print(f"Total de personas: {total_personas}")
```

Total de personas: 20

1. **Contar Personas con Hijos:** Crea un programa que cuente cuántas personas tienen al menos un hijo. Imprime el número de estas personas.

```
# tu pseudocodigo aquí
```

```
# Definir la lista de personas llamada 'people'
# Inicializar una variable llamada 'total_con_hijos' en 0
# Para cada persona en la lista 'people' hacer lo siguiente:
#     Si la cantidad de hijos de la persona es mayor que 0:
#         Incrementar 'total_con_hijos' en 1
# Imprimir el valor de 'total_con_hijos'
```

```
# tu codigo con for aquí
```

```
# Inicializar una variable llamada 'total_con_hijos' en 0
total_con_hijos = 0

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    # Si la cantidad de hijos de la persona es mayor que 0:
    if person["n_kids"] > 0:
        total_con_hijos += 1 # Incrementar 'total_con_hijos' en 1

# Imprimir el valor de 'total_con_hijos'
print(f"Total de personas con al menos un hijo: {total_con_hijos}")
```

Total de personas con al menos un hijo: 5

```
# tu codigo con while aquí
```

```
# Inicializar una variable llamada 'total_con_hijos' en 0
total_con_hijos = 0
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):
    # Si la cantidad de hijos de la persona es mayor que 0:
    if people[index]["n_kids"] > 0:
        total_con_hijos += 1 # Incrementar 'total_con_hijos' en 1
    index += 1 # Incrementar el índice

# Imprimir el valor de 'total_con_hijos'
print(f"Total de personas con al menos un hijo: {total_con_hijos}")
```

Total de personas con al menos un hijo: 5

1. **Contar Hijos Totales:** Escribe un programa que calcule cuántos hijos hay en total entre todas las personas de la lista. Imprime el total.

```
# tu pseudocodigo aquí

# Definir la lista de personas llamada 'people'
# Inicializar una variable llamada 'total_hijos' en 0
# Para cada persona en la lista 'people' hacer lo siguiente:
#     Sumar la cantidad de hijos de la persona a 'total_hijos'
# Imprimir el valor de 'total_hijos'

# tu codigo con for aquí

# Inicializar una variable llamada 'total_hijos' en 0
total_hijos = 0

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    total_hijos += person["n_kids"] # Sumar la cantidad de hijos a
    'total_hijos'

# Imprimir el valor de 'total_hijos'
print(f"Total de hijos entre todas las personas: {total_hijos}")

Total de hijos entre todas las personas: 13

# tu codigo con while aquí

# Inicializar una variable llamada 'total_hijos' en 0
total_hijos = 0
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):
    total_hijos += people[index]["n_kids"] # Sumar la cantidad de
    hijos a 'total_hijos'
    index += 1 # Incrementar el índice

# Imprimir el valor de 'total_hijos'
print(f"Total de hijos entre todas las personas: {total_hijos}")

Total de hijos entre todas las personas: 13
```

1. **Promedio de Edad:** Calcula y muestra la edad promedio de las personas en la lista.

```
# tu pseudocodigo aquí

# Definir la lista de personas llamada 'people'
# Inicializar una variable llamada 'total_edad' en 0
# Para cada persona en la lista 'people' hacer lo siguiente:
#     Sumar la edad de la persona a 'total_edad'
# Calcular el promedio de edad dividiendo 'total_edad' por el número
```

```

de personas
# Imprimir el promedio de edad

# tu codigo con for aquí

# Inicializar una variable llamada 'total_edad' en 0
total_edad = 0

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    total_edad += person["age"] # Sumar la edad a 'total_edad'

# Calcular el promedio de edad
promedio_edad = total_edad / len(people) # Dividir por el número de
personas

# Imprimir el promedio de edad
print(f"El promedio de edad es: {promedio_edad:.2f} años")

El promedio de edad es: 36.29 años

# tu codigo con while aquí

# Inicializar una variable llamada 'total_edad' en 0
total_edad = 0
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):
    total_edad += people[index]["age"] # Sumar la edad a 'total_edad'
    index += 1 # Incrementar el índice

# Calcular el promedio de edad
promedio_edad = total_edad / len(people) # Dividir por el número de
personas

# Imprimir el promedio de edad
print(f"El promedio de edad es: {promedio_edad:.2f} años")

El promedio de edad es: 36.29 años

```

1. **Lista de Personas que Terminan en 'a':** Genera una lista con los nombres de las personas cuyo nombre termina en "a". Imprime esta lista.

```

# tu pseudocodigo aquí

# Definir la lista de personas llamada 'people'
# Inicializar una lista vacía llamada 'nombres_terminan_en_a'
# Para cada persona en la lista 'people' hacer lo siguiente:
#     Si el nombre de la persona termina en 'a':

```

```

#      Añadir el nombre a la lista 'nombres_terminan_en_a'
# Imprimir la lista 'nombres_terminan_en_a'

# tu codigo con for aquí

# Inicializar una lista vacía llamada 'nombres_terminan_en_a'
nombres_terminan_en_a = []

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    # Si el nombre de la persona termina en 'a':
    if person["name"].endswith("a"):
        # Añadir el nombre a la lista 'nombres_terminan_en_a'
        nombres_terminan_en_a.append(person["name"])

# Imprimir la lista 'nombres_terminan_en_a'
print("Personas cuyos nombres terminan en 'a':",
nombres_terminan_en_a)

Personas cuyos nombres terminan en 'a': ['Sonia', 'Lucía', 'Marta']

# tu codigo con while aquí

# Inicializar una lista vacía llamada 'nombres_terminan_en_a'
nombres_terminan_en_a = []
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):
    # Si el nombre de la persona termina en 'a':
    if people[index]["name"].endswith("a"):
        # Añadir el nombre a la lista 'nombres_terminan_en_a'
        nombres_terminan_en_a.append(people[index]["name"])
        index += 1 # Incrementar el índice

# Imprimir la lista 'nombres_terminan_en_a'
print("Personas cuyos nombres terminan en 'a':",
nombres_terminan_en_a)

Personas cuyos nombres terminan en 'a': ['Sonia', 'Lucía', 'Marta']

```

1. **Actualizar Información por Año:** En un año, las personas cuyos nombres terminan en "a" tendrán un niño extra. Crea una nueva lista de diccionarios con la información actualizada de las personas (edad incrementada en 1 y `n_kids` aumentado en 1 para las que terminan en "a"). Imprime esta nueva lista.

```

# tu pseudocodigo aquí

# Definir la lista de personas llamada 'people'
# Inicializar una lista vacía llamada 'nueva_lista'
# Para cada persona en la lista 'people' hacer lo siguiente:

```

```

# Crear un diccionario 'persona_actualizada' con la misma
información que la persona original
# Si el nombre de la persona termina en 'a':
# Incrementar la edad en 1
# Aumentar el número de hijos en 1
# Añadir 'persona_actualizada' a 'nueva_lista'
# Imprimir la nueva lista

# tu código con for aquí

# Inicializar una lista vacía llamada 'nueva_lista'
nueva_lista = []

# Para cada persona en la lista 'people' hacer lo siguiente:
for person in people:
    # Crear un diccionario 'persona_actualizada' con la misma
    información que la persona original
    persona_actualizada = person.copy()

    # Si el nombre de la persona termina en 'a':
    if person["name"].endswith("a"):
        # Incrementar la edad en 1
        persona_actualizada["age"] += 1
        # Aumentar el número de hijos en 1
        persona_actualizada["n_kids"] += 1

    # Añadir 'persona_actualizada' a 'nueva_lista'
    nueva_lista.append(persona_actualizada)

# Imprimir la nueva lista
print("Lista de personas actualizada:")
for persona in nueva_lista:
    print(persona)

Lista de personas actualizada:
{'name': 'Juan', 'age': 34, 'n_kids': 2, 'city': 'Madrid'}
{'name': 'Pepe', 'age': 27, 'n_kids': 0, 'city': 'Barcelona'}
{'name': 'Sonia', 'age': 42, 'n_kids': 2, 'city': 'Valencia'}
{'name': 'Lucía', 'age': 23, 'n_kids': 3, 'city': 'Sevilla'}
{'name': 'Leo', 'age': 55, 'n_kids': 5, 'city': 'Bilbao'}
{'name': 'Marta', 'age': 31, 'n_kids': 4, 'city': 'Zaragoza'}
{'name': 'Carlos', 'age': 45, 'n_kids': 0, 'city': 'Granada'}

# tu código con while aquí

# Inicializar una lista vacía llamada 'nueva_lista'
nueva_lista = []
index = 0 # Inicializar el índice

# Mientras el índice sea menor que la longitud de la lista 'people':
while index < len(people):

```



```
# Crear un diccionario 'persona_actualizada' con la misma información que la persona original
```

```
persona_actualizada = people[index].copy()
```

```
# Si el nombre de la persona termina en 'a':
```

```
if people[index]["name"].endswith("a"):
```

```
    # Incrementar la edad en 1
```

```
    persona_actualizada["age"] += 1
```

```
    # Aumentar el número de hijos en 1
```

```
    persona_actualizada["n_kids"] += 1
```

```
# Añadir 'persona_actualizada' a 'nueva_lista'
```

```
nueva_lista.append(persona_actualizada)
```

```
index += 1 # Incrementar el índice
```

```
# Imprimir la nueva lista
```

```
print("Lista de personas actualizada:")
```

```
for persona in nueva_lista:
```

```
    print(persona)
```

```
Lista de personas actualizada:
```

```
{'name': 'Juan', 'age': 34, 'n_kids': 2, 'city': 'Madrid'}
```

```
{'name': 'Pepe', 'age': 27, 'n_kids': 0, 'city': 'Barcelona'}
```

```
{'name': 'Sonia', 'age': 42, 'n_kids': 2, 'city': 'Valencia'}
```

```
{'name': 'Lucía', 'age': 23, 'n_kids': 3, 'city': 'Sevilla'}
```

```
{'name': 'Leo', 'age': 55, 'n_kids': 5, 'city': 'Bilbao'}
```

```
{'name': 'Marta', 'age': 31, 'n_kids': 4, 'city': 'Zaragoza'}
```

```
{'name': 'Carlos', 'age': 45, 'n_kids': 0, 'city': 'Granada'}
```