

```

...
leer números enteros hasta que el número sea cero y cargar un arreglo con los 6 primeros
número positivos y los 6 primeros números negativos
ingresados. Si no hubiese números negativos o positivos para cargar en el arreglo, mostrar
una leyenda y se termina el ejercicio. De lo contrario:
a) Agregar al arreglo el consecutivo de cada número par (si los hay).
b) Calcular el mayor número positivo y el menor número positivo.
c) Generar otro arreglo con los números múltiplos de 3. Si no los hay mostrar una leyenda.
...

# ----- FUNCIONES -----
def cargar_datos(arr_pos, arr_neg):
    print("\n--- Registro de 6 Positivos y 6 Negativos ---")

    # Primera lectura (Lectura anticipada)
    num_ent = int(input(f"\nIngrese un número (0 para finalizar): "))

    # El ciclo sigue mientras el número no sea 0 Y nos falte completar alguna lista
    while num_ent != 0 and (len(arr_pos) < 6 or len(arr_neg) < 6):
        if num_ent > 0 and len(arr_pos) < 6:
            arr_pos.append(num_ent)
            print(f"Cargado positivo ({len(arr_pos)}/6)")
        elif num_ent < 0 and len(arr_neg) < 6:
            arr_neg.append(num_ent)
            print(f"Cargado negativo ({len(arr_neg)}/6)")
        else:
            print("(!) Aviso: Ese cupo ya está lleno o el número no es válido.")

    # Verificamos si aún necesitamos pedir más números
    if len(arr_pos) < 6 or len(arr_neg) < 6:
        num_ent = int(input(f"\nSiguiente número (0 para finalizar): "))
    else:
        print("\n¡Ambos cupos completados!")

    print("\nCarga finalizada...")
    return arr_pos + arr_neg

# A) Agregar el consecutivo de cada número par
def agregar_consecutivos(arr):
    i = 0
    while i < len(arr):
        if arr[i] % 2 == 0:
            arr.insert(i + 1, arr[i] + 1)
            i += 2
        else:
            i += 1

# B) Calcular mayor y menor positivo
def buscar_extremos_positivos(arr_pos):
    pos_min = 0
    pos_max = 0
    for i in range(1, len(arr_pos)):
        if arr_pos[pos_min] > arr_pos[i]:
            pos_min = i
        if arr_pos[pos_max] < arr_pos[i]:
            pos_max = i
    print(f"\nB) Calcular el número mayor y menor de los positivos.\n")

```

```

    print(f"Mayor positivo: {arr_pos[pos_max]} \nMenor positivo: {arr_pos[pos_min]}")

# C) Generar arreglo con múltiplos de 3
def filtrar_multiplos_3(arr):
    multiplos = []
    for n in arr:
        if n % 3 == 0:
            multiplos.append(n)
    return multiplos

# ----- PROGRAMA PRINCIPAL -----
pos = []
neg = []

arreglo = cargar_datos(pos,neg)
if len(arreglo) == 0:
    print("\nNo se ingresaron datos.")
else:
    print(f"\nArreglo inicial (6 pos y 6 neg):\n \n{arreglo}\n")
    if len(pos) < 6 or len(neg) < 6:
        print("***No se completaron los 6 positivos y 6 negativos.")
    # A
    agregar_consecutivos(arreglo)
    if len(arreglo) > 0:
        print(f"\nA) Con consecutivos de pares: \n\n{arreglo}\n")
    else:
        print("No hay números pares.")
    # B
    buscar_extremos_positivos(pos)
    # C - Múltiplos de 3
    m3 = filtrar_multiplos_3(arreglo)
    if len(m3) > 0:
        print(f"\nC) Múltiplos de 3 encontrados:\n \n{m3}")
    else:
        print("\nC) Leyenda: No se encontraron múltiplos de 3.")
print("\nFin del programa.\n")

```