**Introducción a la programación**

**Tarea #2**

**Elaborado por: Ricardo Artavia Solano**

**PROBLEMA #1:** Realizar la suma de los elementos de una lista de números.

1. **Entender el Problema**

|  |  |  |
| --- | --- | --- |
| **Incógnita:** | **Datos disponibles:** | **Restricción** |
| * Sumatoria de los elementos de una lista de números. | * Lista con números | * La lista debe contener números y debe ser de tipo list. |

**II. Obtener el Plan**

|  |
| --- |
| * Invocar función * Verificar que su argumento sea una lista y que sus elementos sean números. * Ejecutar una función auxiliar. * Sumar recursivamente uno a uno los elementos de esa lista. * Cuando se han recorrido todos los elementos de la lista, desplegar el resultado de la suma.   Identificadores   * Lista de elementos numéricos. * Primer elemento de la lista. * Índices de la lista. |

**III. Aplicar el plan**

|  |
| --- |
| 1. Invocar la función de verificación indicando como su argumento, una lista con elementos numéricos. 2. Leer la lista desde el argumento de la función. 3. Chequear que sea una lista válida y solamente con elementos numéricos, de ser así luego correr la función auxiliar. 4. La función auxiliar obtiene el primer elemento de la lista y lo suma a la invocación de la misma función auxiliar. 5. El proceso se ejecuta recursivamente hasta haber sumado todos los elementos de la lista. 6. Desplegar el resultado de la sumatoria en la pantalla. 7. Terminar proceso |

**IV. Revisar la solución**

|  |  |  |  |
| --- | --- | --- | --- |
| **VARIABLES** | | **Corre Línea** | **Comentario** |
| Lista | Sumatoria |  |
| [1,2,3] |  | 1 | Chequear que sea una lista válida y solamente con elementos numéricos. |
| [1,2,3] |  | 2 | Ejecuta función auxiliar |
| 1 |  | 3 | Obtiene el primer dígito de la lista |
| 1 + func([2,3]) | 1 | 4 | Suma el primer elemento de la lista a la función auxiliar, pero quitando el primer elemento de la lista en su argumento. |
| 1+(2+(3+func[ ])) | 1 | 5 | Repetir el proceso hasta que no haya elementos en la lista para sumar. |
| 1+2+3 | 6 | 6 | Realizar la sumatoria de los elementos |
| 6 | 6 | 7 | Desplegar el resultado en la pantalla. |

**V. Resultado en pantalla**

|  |
| --- |
| >>> suma([1,2,3])  6 |

**Copia del código en estudio:**

#Entradas: lista de números arbitrarios.

#Salidas: sumatoria de todos los elementos de la lista.

#Restricciones: el argumento de la función debe ser del tipo lista

def suma(lista):

if isinstance(lista,list):

return suma\_aux(lista)

else: return "Error, el argumento de la función debe ser una lista"

def suma\_aux(lista):

if lista == []:

return 0

else:

return lista[0] + suma\_aux(lista[1:])

**PROBLEMA #2:** Realizar la suma de los elementos de una lista de números.

1. **Entender el Problema**

|  |  |  |
| --- | --- | --- |
| **Incógnita:** | **Datos disponibles:** | **Restricción** |
| * Sumatoria de los elementos pares de una lista de números. | * Lista con números | * La lista debe contener números y debe ser de tipo “list”. |

**II. Obtener el Plan**

|  |
| --- |
| * Invocar función * Verificar que su argumento sea una lista y que sus elementos sean números. * Ejecutar una función auxiliar. * Sumar recursivamente uno a uno los elementos pares de esa lista. * Cuando se han recorrido todos los elementos de la lista, desplegar el resultado de la suma.   Identificadores   * Lista de elementos numéricos. * Primer elemento de la lista. * Índices de la lista. |

**III. Aplicar el plan**

|  |
| --- |
| 1. Invocar la función de verificación indicando como su argumento, una lista con elementos numéricos. 2. Leer la lista desde el argumento de la función. 3. Chequear que sea una lista válida y solamente con elementos numéricos, de ser así luego correr la función auxiliar. 4. La función auxiliar obtiene el primer elemento de la lista, revisa si este es par y luego lo suma a la función auxiliar, quitando el primer elemento de la lista que esta contiene. 5. El proceso se ejecuta recursivamente hasta haber revisado todos los elementos de la lista. 6. Desplegar el resultado de la sumatoria en la pantalla. 7. Terminar proceso |

**IV. Revisar la solución**

|  |  |  |  |
| --- | --- | --- | --- |
| **VARIABLES** | | **Corre Línea** | **Comentario** |
| Lista | Sumatoria |  |
| [1,2,3,4] |  | 1 | Chequear que sea una lista válida y solamente con elementos numéricos. |
| [1,2,3,4] |  | 2 | Ejecuta función auxiliar |
| 1 |  | 3 | Obtiene el primer dígito de la lista |
| ¿1%2 = 0? | 0 | 4 | Revisa si es par al realizar la división módulo (%) y asegurarse de que el resultado sea 0. |
| return func([2,3,4]) | 0 | 5 | Si no lo es, sigue recorriendo la lista. |
| ¿2%2=0?  2 + func([3,4]) | 2 | 6 | Cuando el dígito es par, se suma a la función nuevamente y esta sigue recorriendo la lista. |
| 2+4 | 6 | 7 | Cuando se hayan recorrido todos los elementos de la lista, realizar la suma de aquellos que son pares. |
| 6 | 6 | 8 | Desplegar el resultado de la sumatoria en la pantalla- |

**V. Resultado en pantalla**

|  |
| --- |
| >>> suma\_pares([1,2,3,4])  6 |

**Copia del código en estudio:**

#Entradas: lista de números arbitrarios.

#Salidas: sumatoria de todos los números pares de la lista.

#Restricciones: el argumento de la función debe ser del tipo lista.

def suma\_par(lista):

if isinstance(lista,list):

return suma\_par\_aux(lista)

else: return "Error, el argumento de la función debe ser una lista"

def suma\_par\_aux(lista):

if lista == [ ]:

return 0

elif lista[0]%2 == 0:

return lista[0] + suma\_par\_aux(lista[1:])

else:

return suma\_par\_aux(lista[1:])