PROBLEMA: Encontrar pares en una lista

**I Entender el Problema**

|  |  |  |
| --- | --- | --- |
| **Incógnita:** | **Datos disponibles:** | **Restricción** |
| * Reconocer pares de lista | * Lista | * El valor ingresado debe ser una lista |

**II Obtener el Plan**

|  |
| --- |
| Verificar que lo que se ingrese sea una lista  Pasar a función de pares  Si el numero en mod 2 da 0 agregarlo a la lista  Si no lo es seguir con el próximo valor de lista |

**III Aplicar el plan**

|  |
| --- |
| 1. Leer lista 2. Isinstance(lista,list)? 3. True: Pasar valor a función de pares 4. False: Devolver error 5. Funcion auxiliar para pares 6. Si lista == []: Devolver [] 7. Si lista[0] % 2 == 0: return [lista[0]] + Funcion(lista[1:]) 8. Else: return función(lista[1:]) 9. Terminar proceso |

**I Revisar la solución Iteracion 1.**

| **VARIABLES** | | | **Corre Línea** | **Comentario** |
| --- | --- | --- | --- | --- |
|  | Lista | Num |
|  | Entrada | Salida |  |  |
|  | [5,2,0,8] | [5,2,0,8] | 1 | Leer lista |
|  | [5,2,0,8]  [5,2,0,8] | [5,2,0,8]  [5,2,0,8] | 2  3  4 | Isinstance(lista,list) ?   1. True: Pasar valor a función de pares 2. False: Devolver error |
|  | [5,2,0,8]  [5,2,0,8]  5  1 | [5,2,0,8]  No hay  No hay  [2,0,8] | 5  6  7  8 | Funcion auxiliar para eliminar valor   1. Si lista == []: Devolver [] 2. Si lista[0] % 2 == 0: return [lista[0]]+ Funcion(lista[1:]) 3. Else: return función(lista[1:]) |
|  |  |  | 9 | Terminar |
|  |  |  |  |  |

**I Revisar la solución Iteracion 1.**

| **VARIABLES** | | | **Corre Línea** | **Comentario** |
| --- | --- | --- | --- | --- |
|  | Lista | Num |
|  | Entrada | Salida |  |  |
|  |  |  | 1 | Leer lista |
|  |  |  | 2  3  4 | Isinstance(lista,list) ?   1. True: Pasar valor a función de pares 2. False: Devolver error |
|  | [2,0,8]  [2,0,8]  2  No hay | [2,0,8]  No hay  0, [2]  No hay | 5  6  7  8 | Funcion auxiliar para eliminar valor   1. Si lista == []: Devolver [] 2. Si lista[0] % 2 == 0: return [lista[0]]+ Funcion(lista[1:]) 3. Else: return función(lista[1:]) |
|  |  |  | 9 | Terminar |
|  |  |  |  |  |

**I Revisar la solución Iteracion 2.**

| **VARIABLES** | | | **Corre Línea** | **Comentario** |
| --- | --- | --- | --- | --- |
|  | Lista | Num |
|  | Entrada | Salida |  |  |
|  |  |  | 1 | Leer lista |
|  |  |  | 2  3  4 | Isinstance(lista,list) ?   1. True: Pasar valor a función de pares 2. False: Devolver error |
|  | [0,8]  [0,8]  0  No hay | ,0,8]  No hay  0, [2,0]  No hay | 5  6  7  8 | Funcion auxiliar para eliminar valor   1. Si lista == []: Devolver [] 2. Si lista[0] % 2 == 0: return [lista[0]]+ Funcion(lista[1:]) 3. Else: return función(lista[1:]) |
|  |  |  | 9 | Terminar |
|  |  |  |  |  |

**I Revisar la solución Iteracion 3.**

| **VARIABLES** | | | **Corre Línea** | **Comentario** |
| --- | --- | --- | --- | --- |
|  | Lista | Num |
|  | Entrada | Salida |  |  |
|  |  |  | 1 | Leer lista |
|  |  |  | 2  3  4 | Isinstance(lista,list) ?   1. True: Pasar valor a función de pares 2. False: Devolver error |
|  | [8]  [8]  8  No hay | [8]  No hay  0, [2,0,8]  No hay | 5  6  7  8 | Funcion auxiliar para eliminar valor   1. Si lista == []: Devolver [] 2. Si lista[0] % 2 == 0: return [lista[0]]+ Funcion(lista[1:]) 3. Else: return función(lista[1:]) |
|  |  | [2,0,8] | 9 | Terminar |
|  |  |  |  |  |

Resultado en pantalla

|  |
| --- |
| [2,0,8] |