

PROBLEMA:

I Entender el Problema

<u>Incógnita:</u>	<u>Datos disponibles:</u>	<u>Restricción</u>
<ul style="list-style-type: none">Concatenar los números primos de una lista	<ul style="list-style-type: none">Lista de longitud n	<ul style="list-style-type: none">Se debe ingresar una lista

II Obtener el Plan

Comprobar si se tiene una lista

Mediante recursividad llamar al primer dígito de la lista y comparar si este es primo o no mediante otro método, si es primo se concatena usando la función `[lista[0]]+ primo(lista[1:])` para así recorrer toda la lista viendo si se tienen primos o no, si no es primo solo se llama a `primo(lista[1:])`.

Parámetros

lista- variable tipo lista-se asume que el valor de la variable es [2,4,3]

III Aplicar el plan

1. Comprobar si el valor de la variable es de tipo lista
2. Si cumple con las condiciones
 - Se pasa al paso 4
3. Si no cumple
 - Se manda mensaje de error
4. Comprobar si `lista[0]` es primo
5. Se utilizan dos parámetros; `lista[0]` y `lista[0]-1` como divisor como condición de parada de la función recursiva se tiene si el divisor es uno o el número es uno por lo tanto se ve que es primo y se devuelve un True
 - Si `num%divisor==0` se cumple no es primo y se devuelve un False
 - Si no pasa lo anterior se devuelve como parámetro `numpri(num,divisor-1)` y se detiene si se recorren los números hasta comprobar que es primo o no.
7. Se compara `numpri(lista[0]lista[0]-1)` con True
 - Si se cumple, el numero de la lista es primo y se concatena usando `[lista[0]]+ primo(lista[1:])`
 - Si no se cumple, se obvia el número y se llama usando la función `primo(lista[1:])`
 - Se repite lo anterior hasta que la lista quede vacía que es la condición de parada y devuelve la lista de primos
 - Terminar proceso

I Revisar la solución

VARIABLES								Corre e Línea	Comentario
Iteración 1		Iteración 2		Iteración 3		Iteración 4			
Lista[0]	primo-return	Lista[0]	primo - return	Lista[0]	primo-return	Lista[0]	primo-return		
								1	Comprobar si el valor de la variable es de tipo lista
								2	Si cumple con las condiciones
									<ul style="list-style-type: none">Se pasa a el paso 4
								3	Si no cumple
									<ul style="list-style-type: none">Se manda mensaje de error
2		4		3				4	Comprobar si lista[0] es primo
								5	Se utilizan dos parámetros; lista[0] y lista[0]-1 como divisor como condición de parada de la función recursiva se tiene si el divisor es uno o el número es uno por lo tanto se ve que es primo y se devuelve un True
									<ul style="list-style-type: none">Si num%divisor== 0 se cumple no es primo y se devuelve un False
									<ul style="list-style-type: none">Si no pasa lo anterior se devuelve como parámetro numpri(num,di

									visor-1) y se detiene si se recorren los números hasta comprobar que es primo o no.
								6	Se compara numpri(lista[0]lista[0]-1) con True
	[2]+[4,3]		[2]+[3]		[2,3]		[2,3]		<ul style="list-style-type: none"> Si se cumple, el numero de la lista es primo y se concatena usando [lista[0]]+primo(lista[1:])
									<ul style="list-style-type: none"> Si no se cumple, se obvia el número y se llama usando la función primo(lista[1:])
									Se repite lo anterior hasta que la lista quede vacía que es la condición de parada y devuelve la lista de primos

Resultado en pantalla

```
listaprimo([2,4,3])
[2,3]
```