**30/07/2021**

*“Si puedes imaginarlo, puedes programarlo”*

Fundamentos de programación

Capítulo # 6

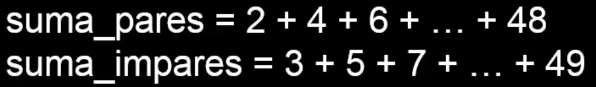
Martinez Rivera Luis Fernando

Ejercicio 2 - Suma de pares e impares comprendidos entre 1 y 50

Descripción del problema:

***Ejercicio 2: Se desea calcular independientemente la suma de los números pares e impares entre 1 y 50.***

***Ejemplo:***

******

***EN ESTE EJERCICIO HUBO 2 POSIBLES SOLUCIONES, LA DEL PROFESOR Y LA MÍA, MI ALGORITMO LO REALICÉ ANTES DE QUE EXPLICARA EL PROBLEMA EL PROFESOR, DE IGUAL FORMA LOS 2 SON CORRECTOS.***

Código fuente (ALGORITMO PROPIO):

// Ejercicio 2: Se desea calcular independientemente la suma

// de los numeros pares e impares entre 1 y 50.

Proceso ejercicio2

Definir i, SUMApares, SUMAimpares Como Entero;

SUMApares<- 0;

SUMAimpares <-0;

Para i<-0 Hasta 50 Con Paso 2 Hacer

SUMApares <- SUMApares + i;

FinPara

Para i<-3 hasta 50 Con Paso 2 Hacer

SUMAimpares <- SUMAimpares + i;

FinPara

Escribir "La suma de los numeros pares entre 1 y 50 da como resultado: ", SUMApares;

Escribir "La suma de los numeros impares entre 1 y 50 da como resultado: ", SUMAimpares;

FinProceso

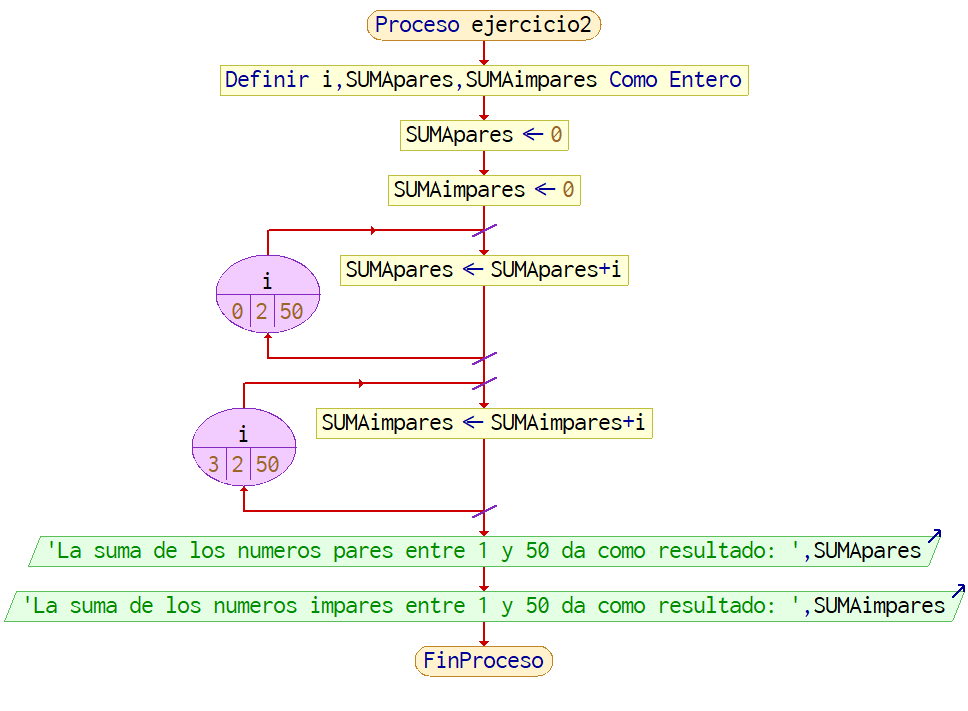
Explicacion del algoritmo:

En este algoritmo lo que utilicé fueron solamente 2 ciclos “para”, mi solución fue que iba a hacer una suma iterativa, pero para poder sumar los números pares e impares lo que hice fue que el iterador (i) siempre fuera ya sea par o impar de acuerdo a la suma que quisiera realizar.

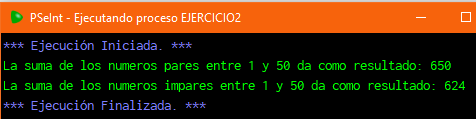
En el primer ciclo lo que hice fue iniciar el iterador en cero e irlo aumentando de 2 en 2 para que así en cada vuelta el iterador siempre fuera un numero PAR y se fuera sumando. Es decir, si el iterador comenzaba en 0 iba a ir aumentando de 2 en 2: 0 – 2 – 4 – 6 – 8 … 50

En el segundo ciclo lo que hice fue iniciar el iterador esta vez en 1 y también irlo incrementando de 2 en 2 para que así en cada vuelta, el iterador fuera un numero IMPAR. Entonces, si el iterador comienza en 1 y aumenta de 2 en 2 esto iría valiendo en cada vuelta: 1 – 3 – 5 – 7 – 9 … 49

Diagrama de flujo:



Capturas de ejecución:



Código Fuente (Ejemplo del profesor):

// Ejercicio 2: Se desea calcular independientemente la suma

// de los numeros pares e impares entre 1 y 50.

Proceso Ejercicio2

Definir i, SUMApar, SUMAimpar como enteros;

SUMApar <- 0;

SUMAimpar <- 0;

Para i<-1 Hasta 50 Hacer

Si i mod 2 = 0 Entonces

SUMApar <- SUMApar + i;

SiNo

SUMAimpar <- SUMAimpar + i;

FinSi

FinPara

Escribir "La suma de los numeros PARES da como resultado: ", SUMApar;

Escribir "La suma de los numeros IMPARES da como resultado: ", SUMAimpar;

FinProceso

Explicación del algoritmo:

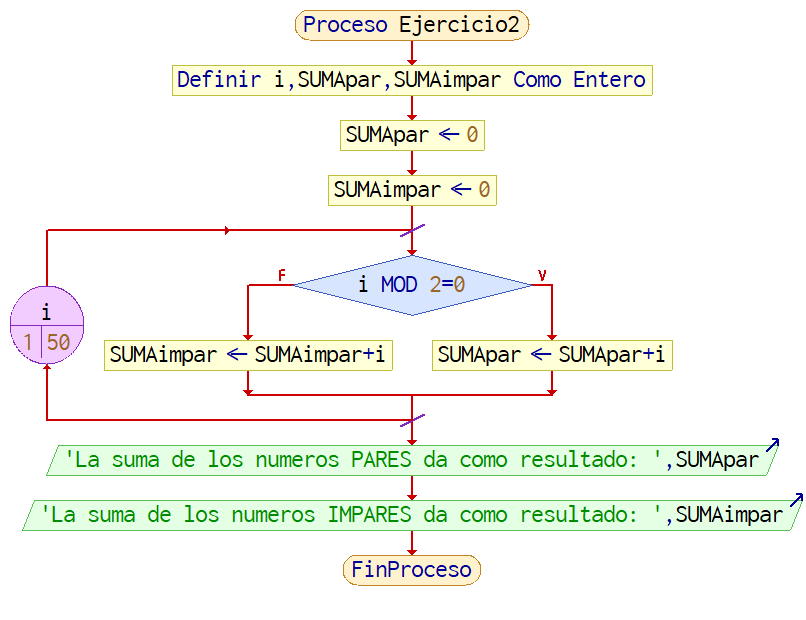
En este ejemplo del profesor se esta usando un condicional para evaluar e identificar al iterador y clasificarlo como numero par o impar.

Se empieza creando un ciclo “para” con un iterador que comienza en 1 y termina en 50 con saltos de 1 en 1, posteriormente en el ciclo se abre una condicional simple y se evalúa el iterador para determinar si es un número par, se hace con el uso de la función modulo o residuo. Se divide al iterador entre 2, si al hacer la operación el modulo es igual a cero (se cumple la condición) quiere decir que es un numero par, de otra forma si al hacer la operación el modulo o residuo no es cero (no se cumple la condicion) quiere decir que es un numero par.

Posteriormente de acuerdo al camino que tome en la condición el iterador se va a ir sumando en su respectivo tipo ya sea par o impar.

Y se vuelve a hacer el ciclo una y otra vez modificando el iterador, en la primera vuelta vale 1, en la segunda 2, 3, 4 , 5 … 50 y en cada una de esas vueltas se evalúa en la condición su valor y se hace lo mismo.

Diagrama de flujo:



Capturas de ejecución:

