A blue and orange square with white letters

Description automatically generated

FUNDAMENTOS DE LA PROGRAMACIÓN PRÁCTICA 2



**GRUPO 40 |** Daniel Atanasov y Raúl Sanz(turno de los miércoles)

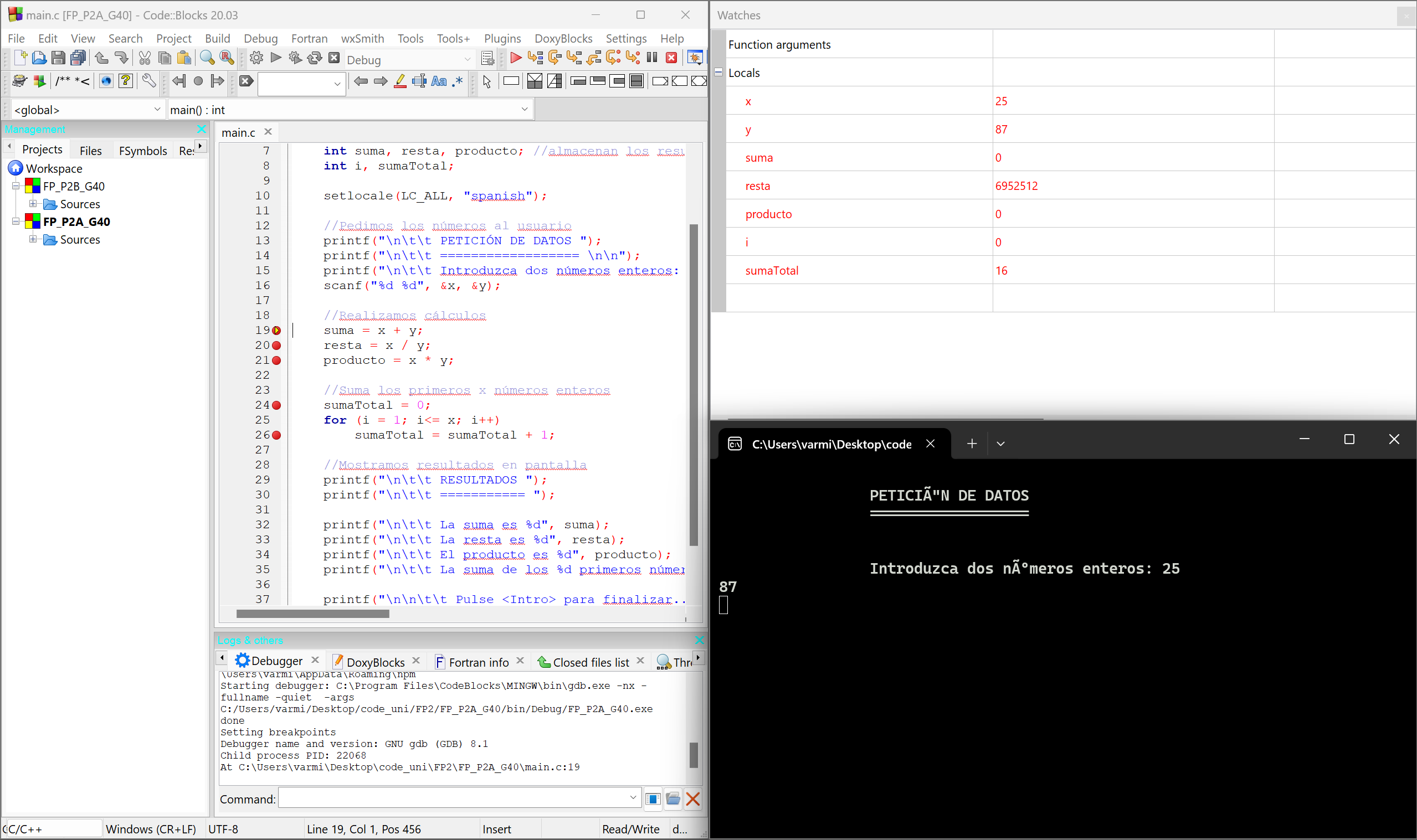
**PRACTICA 2 |** Proceso de Depuración de un Programa y Estructuras de Control

**PROFESORA |** María del Pilar Grande González

ÍNDICE

1. [PARTE 1: Proceso de Depuración de un Programa](#ProyectoA)
   1. Proyecto 2A - Depuración
2. [PARTE 2:](#ProyectoB) Estructuras de Control: Secuencial, Selectiva, Iterativa
   1. Proyecto 2B - Validación de rangos
   2. Proyecto 2C – Menú de opciones
3. PARTE 1: Proceso de Depuración
   1. Proyecto 2A - Depuración

Tras añadir los breakpoints y abrir la ventana de watches, ejecutamos el programa mientras observamos en la ventana de watches la siguiente tabla tras asignar a las variables “x” e “y” lo valores asignados por el usuario:



Según continuamos ejecutando el programa (superando el breakpoint de la línea de código que asigna el valor de la suma), la tabla anterior sufre cambios:

A screenshot of a computer

Description automatically generated

Detectamos el primer error al llegar a la resta, que tiene de valor 0. Para solucionar este problema, hemos sustituidos en el programa el carácter “/” por un “-”, ya que antes de este cambio ejecutaba una división en lugar de una resta.

A screenshot of a computer

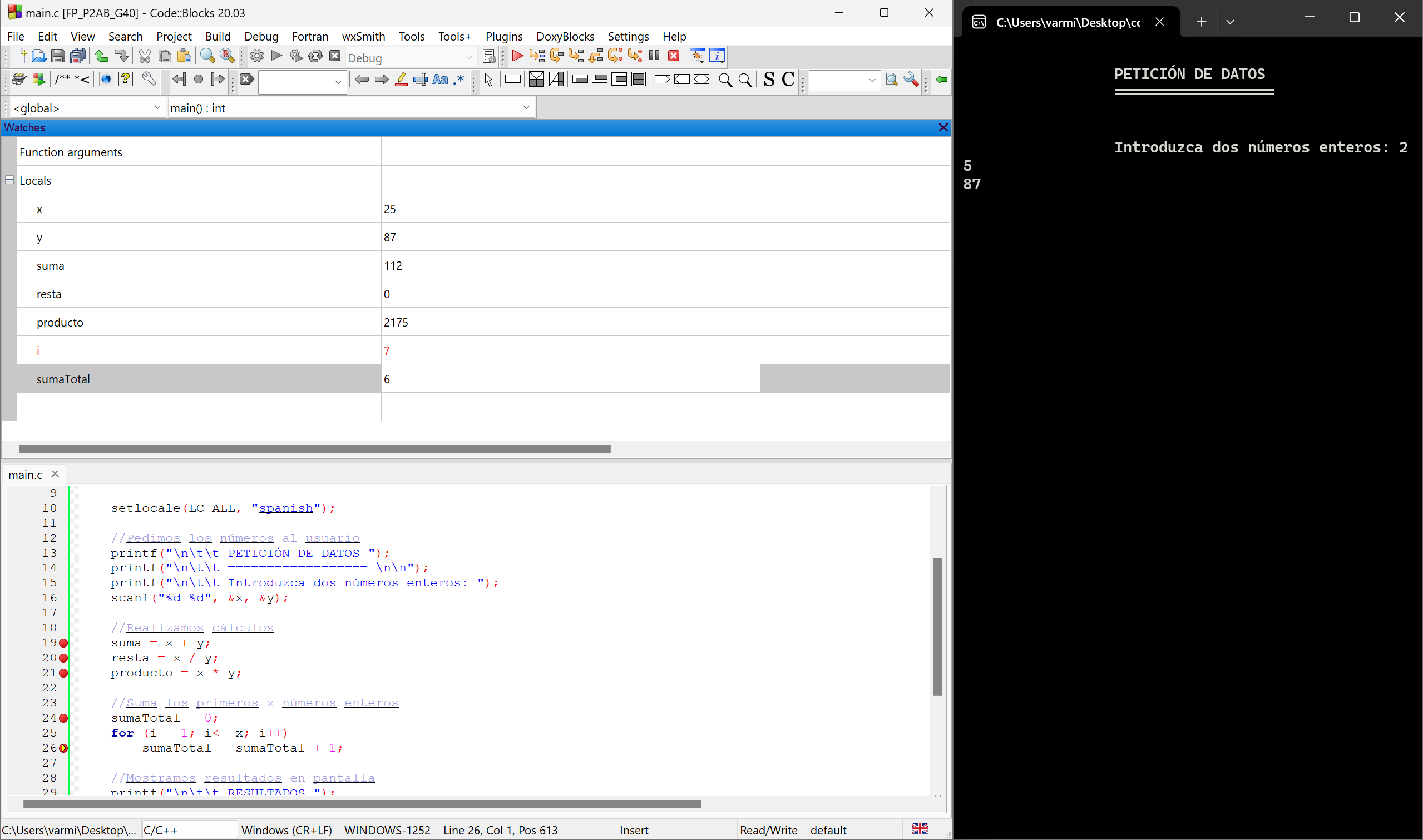
Description automatically generated

A continuación, realiza correctamente el producto:

A screenshot of a computer

Description automatically generated

Por último, llega al bucle, en el que tenemos otro error. Según se ejecuta el bucle, la variable suma 1 por cada ejecución, en lugar de sumar el número correspondiente a la ejecución:

º

Haciendo que al final de la ejecución el valor de “sumaTotal” sea el valor de la variable “x” menos 1:

A screenshot of a computer

Description automatically generated

Para solucionar esto, hemos sustituido el 1 en la suma al final del bucle por el nombre de la variable (i). Al hacer esto, en lugar de sumar 1 cada vez que se ejecuta, se suma el número correspondiente (en esta captura también se puede ver el malfuncionamiento de la resta corregido):

A screenshot of a calendar

Description automatically generated

1. PARTE 2: Estructuras de Control
   1. Proyecto 2B - Validación de rangos

* Análisis del problema

En este proyecto, el programa pide dos números que servirán como rango (mínimo y máximo) y validará que el mínimo es menor que el máximo y que ambos son positivos. Después, pregunta cuantos números quieres comprobar que se encuentran en el rango y solicita que se introduzcan los números.

A continuación, valida los números y muestra en pantalla si el número está o no dentro del rango. Si está dentro del rango, también mostrará todos los números pares menores del número introducido que están dentro del rango.

* Diseño del algoritmo
  + *Diagrama de flujo*
  + *Pseudocódigo*
* Pruebas de ejecución del programa

Adjuntamos, imágenes de la ejecución del programa: