**ACTIVIDAD 3 INTRODUCCIÓN A LA PROGRAMACIÓN**

EJERCICIO 1

* Operadores Aritméticos (Sentencias 1 a 5)

Sentencia 1: (i\_result = x + (y\*2 + x\*3);)

* Orden de Ejecución: Primero, se evalúan las expresiones dentro de los paréntesis. Luego, se realiza la multiplicación y suma de acuerdo con el orden estándar de las operaciones aritméticas.
* Paréntesis: Sí, son necesarios para indicar la prioridad de las operaciones.
* Conversiones de Tipos: Todas las operaciones son enteras, ya que x, y, y las constantes son enteras.
* Resultado: i\_result contendrá el valor 10 (tipo int).

Sentencia 2: (d\_result = x + y\*5 + x\*2;)

* Orden de Ejecución: Primero, se realiza la multiplicación (y\*5 y x\*2), luego se suman los resultados, y finalmente se suma x.
* Paréntesis: No son necesarios en este caso.
* Conversiones de Tipos: d\_result es de tipo double, ya que hay una variable de tipo double (a en lugar de una constante) involucrada en la operación.
* Resultado: d\_result tendrá el valor 18.000000 .

Sentencia 3: (d\_result = x + a\*(y/2.0 + x/2.0);)

* Orden de Ejecución: Primero, se evalúan las expresiones dentro de los paréntesis y se realiza la multiplicación. Luego, se suma x.
* Paréntesis: Sí, son necesarios para indicar la prioridad de las operaciones.
* Conversiones de Tipos: d\_result es de tipo double, ya que hay una variable de tipo double (a) involucrada en la operación.
* Resultado: d\_result tendrá el valor 5.600000.

Sentencia 4: (i\_result = x + a\*(y/2.0 + x/2.0);)

* Orden de Ejecución: Similar a la Sentencia 3, pero el resultado se almacena en una variable entera.
* Paréntesis: Sí, son necesarios para indicar la prioridad de las operaciones.
* Conversiones de Tipos: La operación final se convierte a entero, ya que i\_result es de tipo entero.
* Resultado: i\_result tendrá el valor 5.

Sentencia 5: (d\_result = -a\*(x + y\*(a + b/3.0));)

* Orden de Ejecución: Se evalúan las expresiones dentro de los paréntesis primero, luego se multiplican y suman según el orden estándar de las operaciones aritméticas.
* Paréntesis: Sí, son necesarios para indicar la prioridad de las operaciones.
* Conversiones de Tipos: d\_result es de tipo double, ya que hay una variable de tipo double (a y b) involucrada en la operación.
* Resultado: d\_result tendrá el valor -26.680000.

Sentencia 6:

* Descripción: Multiplica el valor de var1 por 3.2 y asigna el resultado a var1.
* Orden de Ejecución: La multiplicación se realiza primero.
* Resultado: var1 se actualiza con el resultado var1 = 38.

Sentencia 7:

* Descripción: Incrementa el valor de var1 en 1.
* Orden de Ejecución: El incremento se realiza después de la evaluación de la expresión.
* Resultado: var1 se incrementa en 1, var1 = 39.

Sentencia 8:

* Descripción: Asigna el valor actual de var1 a var2 y luego incrementa var1 en 1.
* Orden de Ejecución: La asignación se realiza antes del incremento.
* Resultado: var1 y var2 tendrán el mismo valor después de esta sentencia, pero var1 se incrementará en 1, var1 = 40, var2 = 39 .

Sentencia 9:

* Descripción: Decrementa el valor de var1 en 1 y luego asigna ese nuevo valor a var2.
* Orden de Ejecución: El decremento se realiza antes de la asignación.
* Resultado: var1 se decrementa en 1, y var2 toma el nuevo valor de var1, var1 = 39, var2 = 39.

Sentencia 10:

* Descripción: Divide el valor actual de var2 por sí mismo y asigna el resultado a var2.
* Orden de Ejecución: La división se realiza después de la evaluación de la expresión.
* Resultado: var2 se actualiza con el resultado de la división (que será 1 si var2 no es cero).

Sentencia 11:

* Descripción: Compara si num1 es igual a num2.
* Orden de Evaluación: Se evalúa la igualdad.
* Resultado: resultado tomará el valor de 1 si num1 es igual a num2, y 0 si no lo son,
* resultado = 0.

Sentencia 12:

* Descripción: Evalúa si num1 es mayor o igual a num2, o si num2 es menor que num3.
* Orden de Evaluación: Se evalúa la primera parte de la expresión, y si es falsa, se evalúa la segunda parte.
* Resultado: resultado será 1 si al menos una de las condiciones es verdadera, y 0 si ambas son falsas, resultado = 1.

Sentencia 13:

* Descripción: Evalúa si num2 es menor que 0 y si num3 es mayor o igual a 6.
* Orden de Evaluación: Se evalúa la primera parte de la expresión (num2 < 0), y si es verdadera, se evalúa la segunda parte (num3 >= 6).
* Resultado: resultado será 1 si ambas condiciones son verdaderas, y 0 si al menos una de ellas es falsa, resultado = 1.

Sentencia 14:

* Descripción: Evalúa si num1 es menor o igual a 3 y luego niega el resultado.
* Orden de Evaluación: Se evalúa la desigualdad (num1 <= 3) y luego se aplica la negación (!).
* Resultado: resultado será 1 si num1 no es menor o igual a 3, y 0 si lo es, resultado = 1.

Sentencia 15:

* Descripción: Desplaza a la izquierda los bits de i en j posiciones.
* Resultado: res\_bits contendrá el resultado de desplazar los bits de i hacia la izquierda en j posiciones, res\_bits = 40.

Sentencia 16:

* Descripción: Desplaza a la derecha los bits de i en j posiciones, llenando con ceros en el lado izquierdo.
* Resultado: res\_bits contendrá el resultado de desplazar los bits de i hacia la derecha en j posiciones, res\_bits = 0.

Sentencia 17:

* Descripción: Asigna la dirección de memoria de valor1 a la variable puntero p.
* Resultado: p contiene la dirección de memoria de valor1, p = cad1cacc.

Sentencia 18:

* Descripción: Asigna el valor almacenado en la dirección de memoria apuntada por p a la variable valor2.
* Resultado: valor2 toma el valor contenido en la dirección de memoria apuntada por p,
* valor2 = 7.

Sentencia 19:

* Descripción: Asigna la dirección de memoria de valor2 a la variable puntero q.
* Resultado: q contiene la dirección de memoria de valor2, q = cad1cac8.

Sentencia 20:

* Descripción: Asigna el valor almacenado en la dirección de memoria apuntada por q a la variable valor3.
* Resultado: valor3 toma el valor contenido en la dirección de memoria apuntada por q,
* valor3 = 7.

RESULTADOS COMPILACIÓN

(primera compilación en online C compiler, segunda como archivo .c en terminal)

1ª Compilación

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

2ª Compilación

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Los resultados concuerdan, y el programa se ha compilado y ejecutado correctamente.

EJERCICIO 2

Compilación realizada en terminal

Texto

Descripción generada automáticamente

Como se puede ver el programa imprime a la perfección los números primos obviando los compuestos del 0 al 100.