

Tarea 7

1. Suponga que $i = 1$, $j = 2$, $k = 3$, $m = 2$. ¿Qué es lo que imprime cada uno de los enunciados siguientes y por qué?

- a) `printf("%d", i == 1);`
- b) `printf("%d", j == 3);`
- c) `printf("%d", i >= 1 && j > 4);`
- d) `printf("%d", m <= 99 && k < m);`
- e) `printf("%d", j >= i || k == m);`
- f) `printf("%d", k+m < j || 3 - j >= k);`
- g) `printf("%d", !m);`
- h) `printf("%d", !(j - m));`
- i) `printf("%d", !(k < m));`
- j) `printf("%d", !(j > k));`

2. Encuentra el error en cada uno de los segmentos siguientes de programa y explica cómo puede corregirse dicho error. Supone que se declaró el prototipo de las siguientes funciones de forma correcta.

```
int sum(int x, int y)
{
    int resultado;
    return = x + y;
}
```

```
void f(float a)
{
    float a;
    printf("%f", a);
}
```

```
void producto()
{
    int a, b, resultado;
    printf("Ingresa dos números: ");
    scanf("%d %d", &a,&b);
    resultado = a * b;
    printf("El producto es: %d\n", resultado);
    return resultado;
}
```

3. Escribe un programa que contenga una función que regrese el más pequeño de tres números de tipo **float**. Utiliza esta función en la **función principal**. A continuación se muestra un ejemplo:

Entrada	Salida
2.4, -0.5, 456.2	-0.5

4. Escribe una función que regrese la longitud de la hipotenusa de un triángulo rectángulo, cuando son conocidos los otros dos lados. La función debe tomar dos argumentos de tipo **float** y regresar la hipotenusa también de tipo **float**. Utiliza esta función en un programa para determinar la longitud de la hipotenusa de los triángulos siguientes.

Triángulo	Lado 1	Lado 2	hipotenusa
1	3.0	4.0	Valor calculado
2	5.5	12.0	Valor calculado
3	8.1	15.0	Valor calculado

5. **RETO:** Escribe un programa que contenga una función que tome un valor entero (número) y la cantidad de dígitos que tiene éste y que imprima el número con sus dígitos invertidos. Utiliza esta función en la **función principal**. Tips:
- Toma las ideas del ejercicio hecho en clase, en el que hacíamos divisiones y usábamos el operador módulo.
 - Es posible definir arreglos muy grandes, digamos `int f[100]`, puedes definir un arreglo así para ir guardando los dígitos.

A continuación se muestra un ejemplo:

Entrada	Salida
7631, 4	1367
634, 3	436
87462, 5	26478