

## Tarea 2 - Programación y algoritmos

### Giovanni Gamaliel López Padilla

## Problema 1

¿Cuál es la diferencia entre el `format %i` y `%d`? Dé un ejemplo. ¿C es la diferencia entre la declaración `bool` y `_Bool` en C?

La diferencia entre el formato

## Problema 2

¿Qué pasa si al leer un entero con `scanf()`, el usuario teclea el número seguido con una letra? Ejem: `67f`, ¿como explica el resultado? Enseguida de esta instrucción, añade ahora la lectura de un carácter, ¿Qué pasa y como explica este comportamiento?

La razón por la cual se realiza la lectura del número unicamente es debido a que la función `scanf` se le pasa el formato de lo que debería de leer, es por ello que asigna el valor numérico a la variable que se le dio.

El carácter se queda guardado en alguna variable interna, es por ello que al realizar la petición de un carácter es asignada a esta variable de carácter.

## Problema 3

Programa que realice una operación aritmética especificada entre dos fracciones. La entrada debe ser de la forma: `a/b×c/b`, donde  $\times \in \{+, -, *, /\}$ .

El programa se encuentra en la carpeta [Problema\\_3](#). La manera de ejecutar el programa es con la siguiente línea:

```
gcc -Wall -o main.out main.c -std=c11
```

## Problema 4

Programa que imprima un número entero dado de `n` dígitos al revés. Ejem, entrada: `79373`, salida: `37397`.

La estructura del código es la siguiente:

```
n = 79373
size = obtain_number_of_digits(n)
number_list=integer_to_list(n)
# number_list = [7, 9, 3, 7, 3]
n_flip = 0
for i=0,1,2,...,size-1:
    n_flip+=number_list[size-i-1]*10^(i)
print(n_flip)

> output: 37397
```

La manera de obtener el número de dígitos en un número entero es contar las veces que se puede dividir el número entre 10 hasta obtener un número menor a 1. Ya obtenido esto se creo

una lista donde cada elemento contiene un dígito del número que queremos escribir al revés. La sustracción de cada dígito se obtuvo con la ecuación 1.

$$list[i] = number * 10^{i-n} - \sum_{j=0}^{i-1} list[j] * 10^{i-j} \quad \forall i = 1, 2, \dots \quad (1)$$

donde  $i$  es la posición del dígito que queremos sustraer,  $n$  es el número total de dígitos que tenemos,  $number$  es el número entero que queremos escribir al revés y  $list$  es la lista que contiene a cada dígito. Al ser esta una ecuación recursiva, se necesita saber su valor inicial ( $list[0]$ ). El calculo del valor inicial esta descrito en la ecuación 2.

$$list[0] = number * 10^{-n} \quad (2)$$

El valor del número al revés se calculó con la ecuación 3.

$$number_{flip} = \sum_{i=0}^n list[n-i] * 10^{n-i} \quad (3)$$

El programa se encuentra en la carpeta [Problema\\_4](#). La manera de ejecutar el programa es con la siguiente linea:

```
gcc -Wall -o main.out main.c -std=c11 -lm
```

## Problema 5

Programa que evalué la siguiente expresión. Debe de pedir el número de términos a evaluar.

$$e^{-x^2} = \sum_{i=0}^{\infty} \frac{(-x^2)^i}{i!} = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!} x^{2i}$$

Problema terminado

## Problema 6

Programa que convierta un número decimal a cualquier base.

## Problema 7

Programa que acepte un fracción del tipo  $a/b$  tal que  $(a,b) \in \text{enteros}$ , y la reduzca a sus términos mínimos. Ejem:  $6/24 \rightarrow 1/4$ . Debe encontrar el GCD.

El programa se encuentra en la carpeta [Problema\\_7](#). La manera de ejecutar el programa es con la siguiente linea:

```
gcc -Wall -o main.out main.c -std=c11
```