

TRABAJO PRÁCTICO N° 3

Deben entregarse por grupos en la entrega correspondiente vía Campus los archivos.c correspondientes a los ejercicios indicados.

1. Predecir la salida del siguiente código:

```
#include <stdio.h>

int a = 3;

void fun1(void)
{
    printf("%d\n",a++);
}

void fun2(void)
{
    int a = -2;
    printf("%d\n",a++);
}

int main()
{
    int a = 10;
    printf("%d\n",a++);
    fun1();
    fun2();
    fun1();
    fun2();
    printf("%d\n",a++);
    return 0;
}
```

Compilar el programa y verificar que realice lo predicho.

2. Sea un programa, llamado `ej1`, se ejecuta con el comando:

```
user@pc :~$ ./ej1 < test.txt
```

El contenido del archivo `test.txt` es:

```
abcdefghi
ihgfedcba
```

Predecir la salida del programa ej1, si el código es el siguiente:

```
#include <stdio.h>
int main(void)
{
    int c;
    while ((c = getchar()) != 'e' )
    {
        putchar(c+'A'-'a');
    }
    return 0;
}
```

Compilar el programa y verificar que realice lo predicho.

3. Escribir un programa que imprima en pantalla los caracteres recibidos por teclado. El programa debe terminar cuando se recibe el caracter 'q' ó 'Q'.
4. Escribir una función **double** *cuadrado(double x)* que devuelva el valor ingresado elevado al cuadrado.
5. Escribir un programa que reciba un número por teclado e imprima dicho número elevado al cuadrado (usar la función del ejercicio anterior para realizar la cuenta). Se debe validar completamente el número ingresado (es decir, indicar al usuario que hubo un error si el formato del número ingresado es incorrecto), y se debe poder ingresar cualquier número real (en base decimal, positivo y negativo). **No utilizar la función scanf().**
6. Reescribir el programa del ejercicio 3 del TP2 (evaluación de temperatura), permitiendo al usuario ingresar la temperatura por teclado, y encapsulando el código que imprime el mensaje según la temperatura dentro de una función. **Pista:** se puede reutilizar la función que recibe números que escribieron en el ejercicio anterior (en este ejercicio y en los siguientes).
7. Reescribir el programa del ejercicio 5 del TP2 (serie de Taylor), modularizando el programa en subfunciones específicas.
8. Reescribir el programa del ejercicio 6 del TP2 (evaluación de números primos), permitiendo al usuario ingresar el número que se desea evaluar por teclado, y encapsulando el código que determina si un número es primo dentro de una función.
9. Usar redireccionamiento de la entrada estándar para ingresarle al programa del ejercicio anterior el número a evaluar desde un archivo (p.ej. primo.txt).

10. **[ENTREGAR]** Escribir un programa que reciba del usuario, por teclado, una fecha inicial y una fecha final, e imprima en pantalla la cantidad de días que hay comprendidos entre ellas. El programa debe validar por completo las fechas recibidas, es decir, debe indicar al usuario que hubo un error si recibe una fecha mal formateada o imposible (ej. 30 de febrero). El usuario deberá ingresar 6 valores: día, mes y año de la fecha inicial y día, mes y año de la fecha final, según se lo indique el programa (es decir, el programa indica al usuario "ingrese el día inicial", luego "ingrese el mes inicial" etc.). Debe estar correctamente modularizado en subfunciones específicas. **No utilizar la función `scanf()`.**
- a. Considerar que todos los años múltiplo de 4 son bisiestos, salvo que sean múltiplos de 100, en cuyo caso también deben ser múltiplo de 400.¹
 - b. Al momento de testear el programa, ser exhaustivo en cuanto a las pruebas realizadas. Es decir, probar varias condiciones posibles: dos fechas en el mismo mes, dos fechas en distinto mes y mismo año, dos fechas en distinto año pero el mismo mes, etc.

Consejo: Concentrarse en escribir funciones cortas que implementen los distintos bloques del programa, en vez de hacer funciones largas y complejas que realizan gran parte del cálculo dentro de ellas. Pensar como se va a modularizar el programa antes de sentarse a escribir la primer línea.

¹ https://es.wikipedia.org/wiki/Año_bisiesto