

Clase de Laboratorio

1. Implementar este código en el compilador y documentarlo en función de lo que realiza.

```

#include <stdlib.h>
#include <stdio.h>

void main (void)
{
    int cantIBlancos = 9;
    int cantIAsteriscos = 1;
    int i, j, k;

    for (k = 1; k <= 10; k++)
    {
        for (j = 1; j <= cantIAsteriscos; j++)
            printf ("*");
        for (i = 1; i <= cantIBlancos; i++)
            printf ("-");
        printf ("\n");

        cantIAsteriscos++;
        cantIBlancos--;
    }

    printf ("\n");
    system ("pause");
}

```

Sobre el mismo código cambiando solamente:

- La inicialización de las variables.
- El incremento o decremento de los for.
- El orden en que se ejecutan los for.

Crear 3 programas más que muestren las siguientes salidas:

-----*	*****	*****
-----**	_*****	*****_
-----***	__*****	*****__
-----****	___*****	*****___
-----*****	____*****	*****____
-----*****	-----*****	*****-----
____*****	-----****	****_-----
*****	-----***	***-----
*****	-----**	**-----
*****	-----*	*_-----

2. Realizar un programa en C que solicite al usuario el ingreso de un número entero (long) y muestre en pantalla:

- La cantidad de dígitos que tiene el mismo.
- El mayor de dichos dígitos.

Realizar pruebas tanto con números positivos como negativos.

3. Considere el siguiente programa:

```
#include <stdlib.h>
#include <stdio.h>

int main (void)
{
    int i, numero;
    int bandera = 0;

    printf ("Ingrese un numero entero: ");
    scanf ("%i", &numero);

    for (i = 2; i < numero; i++)
        if ((numero % i) == 0)
            bandera++;

    printf ("\n");
    system ("pause");
    return 0;
}
```

- Complete el programa para que muestre al usuario si el número ingresado es primo o no lo es.
- Realice pruebas para determinar si falla ante alguna entrada válida.
- Si encuentra algún error corríjalo.
- Modifique el programa para que determine los números primos que se encuentran entre el 1 y el 1000.

4. Modifique el ejercicio 1 para implementarlo con otras estructuras de control (no utilizar "for").

Ejercicios Complementarios

5. Realice un programa para el calculo del salario que debe ser abonado a un trabajador por el trabajo realizado en una semana, siendo que existen tres tipos de trabajadores con los siguientes precios por hora de trabajo:

Tipo empleado	Precio hora normal	Precio hora extra
Oficial	\$30,50	\$40,00
Peón	\$20,50	\$27,00
Especializado	\$35,00	\$48,00

El programa debe pedir el ingreso del tipo de empleado y la cantidad de horas que trabajó en la semana e informar el salario que le corresponde. Hasta 40 horas de trabajo semanal se toman como horas normales y el resto se toman como horas extras.

6. Realice un programa que encuentre y muestre todos los números de 4 cifras que cumplen la condición de que la suma de las cifras de orden impar es igual a la suma de las cifras de orden par. Ejemplo: 3245 => 3+4=2+5.

Ayuda: para obtener el último dígito de un número se puede utilizar el operador módulo o resto de la división (%) con dicho número y el 10. Ejemplo: 3245 % 10 => 5

7. Realice un programa que lea las puntuaciones de exámenes de un curso en un rango entre 0 y 100, luego del ingreso de cada puntuación el programa preguntara al usuario si desea seguir ingresando datos. Cuando se termine el ingreso, calcule e informe, de acuerdo a la siguiente tabla, para cada nota la cantidad de alumnos que la obtuvieron y el porcentaje sobre el total.

Rango	Nota
90-100	A
80-89	B
70-79	C
60-69	D
0-59	E

8. Realice un programa que imprima una figura en diamante como la siguiente:

```
      *
    * * *
  * * * * *
* * * * * *
  * * * * *
    * * *
      *
```

Donde la cantidad de filas es especificada por un número impar leído desde teclado en el rango 1 y 19 (validar)