



## Programación I

### Listado de práctica

1. Escriba un programa que pida un número entero  $n$  e imprima “primo” si el número es primo o “compuesto” en caso contrario. (**Nota:** Basta con verificar los números hasta  $\sqrt{n}$ .)
2. Observe el siguiente patrón de impresión de 8 niveles. Escriba un programa que pida un número entero  $n$  e imprima el mismo patrón on  $n$  niveles.

```
#
++
###
++++
#####
++++++
#####
+++++++
```

3. Observe el siguiente patrón de impresión de 8 niveles. Escriba un programa que pida un número entero  $n$  e imprima el mismo patrón on  $n$  niveles.

```
# #####
++ ++++++
### #####
++++ +++++
##### #####
++++++ +++
##### ##
+++++++ +
```

4. Escriba un programa que pida dos números enteros  $m$  y  $n$ , e imprima el resultado de la combinatoria  $\binom{m}{n}$ .
5. Escriba un programa que pida 3 coordenadas  $(x, y)$  e imprima “contenido” si la tercera coordenada está contenida en el rectángulo delimitado por las primeras dos coordenadas. En caso contrario, imprima “no contenido”.
6. En algunas oficinas del gobierno pagan horas extra a los burócratas, además del salario correspondiente. Escriba un programa en C que permita calcular la cantidad a pagar a un trabajador por concepto de horas extra trabajadas. Las horas extra se pagan según la categoría a la que pertenece cada trabajador y se calculan en función de la siguiente tabla. Cada trabajador puede tener como máximo 30 horas extra, si tienen más, sólo se les pagarán las primeras 30. Los trabajadores con categoría 4 o mayor a 4 no pueden recibir este beneficio.

Categoría trabajador	Valor hora extra
1	\$20.000
2	\$23.000
3	\$25.000



7. Escriba un programa que pida dos números enteros  $n$  e  $y$ , e imprima el resultado de las funciones  $f(1)$ ,  $f(2)$ ,  $f(3)$ , ...,  $f(n)$ , donde  $f(x)$  se define de la siguiente forma:

$$f(x) = \begin{cases} y^2 + 15 & \text{si } 0 < x \leq 15 \\ y^3 - y^2 + 12 & \text{si } 15 < x \leq 30 \\ 4y^3/y^2 + 8 & \text{si } 30 < x \leq 60 \\ 0 & \text{si en otro caso} \end{cases} \quad (1)$$

8. Escriba un programa que pida un número entero  $n$ , que luego genere dos arreglos ordenados de largo  $n$ ,  $A$  y  $B$ , con valores al azar, y finalmente genere un arreglo ordenado  $C$  que sea el resultante de la unión de  $A$  y  $B$ .

Por ejemplo, si los arreglos generados son  $A = [1, 5, 9, 15]$  y  $B = [0, 6, 7, 8]$ , el programa debe generar el arreglo  $C = [0, 1, 5, 6, 7, 8, 9, 15]$ .

9. Escriba un programa que pida un número entero positivo  $n$  y que luego pida suficientes números que serán almacenados en una matriz triangular superior de  $n \times n$ .
10. Escriba un programa que pida una cadena de caracteres y que reporte la cantidad de caracteres en mayúscula, minúscula, dígitos y espacios. Note que la cadena a ingresar debe permitir espacios.
11. Provea una solución al problema 8, pero esta vez usando exclusivamente memoria dinámica.
12. Escriba la función `void traspuesta(int n, int m, int M[n][m], int T[m][n])` que reciba la matriz  $M$  de  $n$  filas y  $m$  columnas, y compute su traspuesta, la cual será almacenada en la matriz  $T$ .
13. Escriba la función `void mult(int n, int m, int l, int M[n][m], int T[m][l], int R[n][l])`, la cual computará la multiplicación de las matrices  $M$  y  $T$ . La matriz resultante quedará almacenada en la matriz  $R$ .
14. Escriba la función `int identidad(int n, int m, int M[n][m])` que reciba la matriz  $M$  de  $n$  filas y  $m$  columnas, y retorne 1 si la matriz  $M$  es la matriz identidad y 0 en caso contrario.
15. Escriba la función `int ortogonal(int n, int m, int M[n][m])` que reciba la matriz  $M$  de  $n$  filas y  $m$  columnas, y retorne 1 si la matriz  $M$  es ortogonal y 0 en caso contrario. Para responder esta pregunta **debe** utilizar las funciones resultantes de las preguntas 12, 13 y 14.

**Nota:** Una matriz cuadrada  $A$  es ortogonal si se cumple  $A \times A^T = I$ , donde  $A^T$  es la matriz traspuesta de  $A$  e  $I$  es la matriz identidad.

16. Implemente las siguientes 3 estructuras:

- Estructura `video_juego`, la cual estará conformada por los campos nombre del video juego, año de lanzamiento y consolas o plataformas.
- Estructura `compra`, la cual estará conformada por los campos fecha de la compra, monto de la compra y puntero al video juego comprado.
- Estructura `tienda`, la cual estará conformada por los campos nombre de la tienda, dirección, arreglo de video juegos y arreglo de compras.

17. Utilizando las estructuras del ejercicio 16, implemente las funciones para agregar video juegos a la tienda, registrar una nueva compra, imprimir registros de video juegos, imprimir registros de compras y eliminar un video juego.

**Tips:** Utilice memoria dinámica.

18. Escriba una función recursiva que implemente la función `itoa`: Convertir un entero en una cadena de caracteres la cual estará conformada por los dígitos que conforman dicho número.

Por ejemplo, dado el entero 4570, la implementación de la función recursiva debe generar la cadena "4570".

19. La búsqueda binaria es un algoritmo que permite realizar búsquedas rápidas en un arreglo ordenado. Investigue su funcionamiento e implemente una versión recursiva del algoritmo.