



**Examen parcial 06/05/2019**

**CC112**

**Ciclo: 2019-1**

**Normas:**

1. El alumno entregará esta hoja de examen debidamente llenada con sus datos.
2. La solución de la prueba se guardarán en **Escritorio**, carpeta: **ApellidoNombreCodigo** (sin espacios en blanco), la pregunta **n** se guardará en el archivo: **n.c** ( $n = 1, 2, \dots$ ).
3. No se permite: El uso de celulares, internet, USB, ingresar después de 15 min. de iniciado el examen ni salir antes de la hora de finalización.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos : \_\_\_\_\_ Nombres : \_\_\_\_\_

Sección : \_\_\_\_ Grupo: \_\_\_\_

1. [5 pts.] **Usando estructura selectiva**, escriba un **programa** que ayude a una empresa a incrementar los salarios de los trabajadores, según los rangos siguientes:

RANGO DE SUELDO	AUMENTO	// mostrar todo en el monitor
a: De \$ 1 a \$ 9000	20%	
b: De \$ 9000 a \$15000	10%	
c: De \$15000 a \$20000	5%	
d: Más de \$20 000	3%	

Indique su rango de sueldo (a,b,c o d): a  
Ingrese su sueldo actual: 5000  
Su nuevo sueldo sera: 6000

2. [5 pts.] **Usando bucles anidados**, escriba un programa que muestra el primer número perfecto mayor que 100. Un número es perfecto cuando la suma de sus divisores excepto él mismo es igual al propio número. Por ejemplo 6 es perfecto ya que sus divisores son 1, 2, 3 y suman 6  
Salida en el monitor:

Primer numero perfecto mayor que 100 = 496

3. [5 pts.] La **criba de Eratóstenes**, es uno de los algoritmos más antiguos que lo debemos a un astrónomo, geógrafo, matemático y filósofo de la antigua Grecia, permite hallar todos los números primos menores que un número natural dado  $N$ . Se forma un arreglo de una dimensión con todos los números naturales comprendidos entre 2 y  $N$ , se *inicia en 2* y se van tachando los números que no son primos de la siguiente manera: cuando se encuentra un número entero que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos. El proceso termina cuando el cuadrado del mayor número confirmado como primo es mayor que  $N$ . Programe dicho algoritmo para  $N=100$ .

Sugerencia de diseño:

Defina: char criba[N];  
Inicie: criba[0] = 0; criba[1] = 0; el resto = 1;  
Asigne: 0 a todos los elementos no primos.

4. [5 ptos.] **Torres de Hanoi**, es un juego que consiste en tres varillas verticales: A, B y C, en A están apiladas un número de discos, de diámetros diferentes, ordenados de mayor a menor (el de mayor diámetro abajo). B y C están vacías. El juego consiste en pasar todos los discos de la varilla A a C, la varilla B es auxiliar.

Reglas:

- Sólo se puede mover un disco cada vez.
- Un disco de mayor tamaño no se puede colocar encima de uno más pequeño.
- Sólo se puede mover el disco que se encuentre en la parte superior de cada varilla.

Escriba un programa que permita ingresar el número de discos y muestre todos los movimientos que se debe realizar, ejemplo de salidas en el monitor para distintos N:

N = 1	N = 2	N = 3
A->C	A->B	A->C
	A->C	A->B
	B->C	C->B
		A->C
		B->A
		B->C
		A->C

**Notas:**

- 1) La acción "mover disco de A a B", en el mundo real, es ejecuta en el programa (mundo virtual) por printf("...", 'A', 'B'): A->B. Como puede ver, ya estamos cerca de la robótica.
- 2) No se asuste con este problema, se resuelve muy rápidamente con una función recursiva, siguiendo un algoritmo, tal y como se hace para encontrar el máximo común divisor de dos números siguiendo el algoritmo de Euclides.
- 3) En un problema complejo, el programador debe seguir las especificaciones del diseñador; aunque no entienda la lógica de la solución (Es una regla para trabajar en equipo). Para validar el programa debe usar la matriz de casos de prueba (en este caso para N=1, 2 y 3).

**Diseño:**

Defina: char com='A', aux='B', fin='C';

Llame: hanoi(N, com, aux, fin); // "mueve" N discos de com a fin utilizando aux  
// como auxiliar

```
hanoi(...){
    if(n==1) printf("%c->%c",com,fin); // "mueve" disco más pequeño de com a fin
    else {
        hanoi(...); // "mueve" N-1 discos de com a aux, utilizando fin
                    // como si fuera aux
        printf("\n%c->%c\n",com,fin); // "mueve" el disco más grande de com a fin
        hanoi(...); // "mueve" N-1 discos de aux a fin, utilizando com
                    // como si fuera aux
    }
}
```