



Cuarta Práctica Calificada

CC102-AB

Ciclo: 2017-2

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 ptos.] Escriba un programa que solicite un entero $n > 0$, luego solicite n enteros que los alojará en memoria dinámica, finalmente los imprime, ejemplo de salida:
Cuantos numeros va a ingresar?: 2
Número 1: 10
Número 2: 20
Números ingresados:
Número 1: 10
Número 2: 20
2. [5 ptos.] Escriba un programa que reserve memoria dinámica para 3 matrices de enteros: $A[2][3]$, $B[3][2]$ y $C[2][2]$. Asigne datos: elementos de $A = 1$, elementos de $B = 2$, $C = A \times B$. Imprima A , B y C , ejemplo de salida:
Matriz A
1 1 1
1 1 1
Matriz B
2 2
2 2
2 2
Matriz C
6 6
6 6

Sugerencia: Complete el programa:

```
#include<stdio.h>
#include<stdlib.h>
int * espacio(int m, int n);
void asigna(int *pa, int m, int n, int k);
void por(int *pa, int *pb, int *pc, int k, int m, int n);
void imp(int *p, int m, int n);
void main(void){
    int *pa, *pb, *pc;
    int k=2, m=3, n=2;
        // complete las operaciones
}
int * espacio(int m, int n){
    // reserva espacio para m*n enteros
}
```

```

void asigna(int *pa, int m, int n, int k){
    // asigna el valor k a todos los elementos del espacio de m*n elementos
}
void por(int *pa, int *pb, int *pc, int k, int m, int n){        // C = A x B
    int i, j, l;
    int suma;
    for(i=0; i<k; i++){
        for(j=0; j<n; j++){
            suma=0;
            for(l=0; l<m; l++) suma += pa[i*m+l] * pb[l*n+j];
            *pc++ = suma;
        }
    }
}
void imp(int *p, int m, int n){
    // imprimir un espacio de m*n enteros en formato de matriz
}

```

3. [5 ptos.] El sueldo mensual de un vendedor de autos es el sueldo base más su comisión; dicha comisión surge del producto de la tasa del mes por la cantidad de autos vendidos. Implemente una estructura "vendedor" cuyos miembros sean una cadena de caracteres (para almacenar el nombre) y un tipo de dato numérico (para almacenar el sueldo base). La tasa es de 200 soles/auto vendido y se vende menos de 10 autos/vendedor, simule este dato. Calcule el sueldo de dos vendedores estrella: ingrese el nombre de pila y el sueldo base de cada vendedor; finalmente calcule el total de pagos. Ejemplo de salida:

Datos de vendedores[1]:

Nombre : **AA**

Sueldo base: **200**

Datos de vendedores[2]:

Nombre : **BB**

Sueldo base: **300**

Nombre	Base	Comisión	Pago
AA	200.00	1600.00	1800.00
BB	300.00	800.00	1100.00
Total pagos:			2900.00

4. [5 ptos.] Se tiene una tabla de profesores con los siguientes datos:

Nombre	Sueldo	Edad	// estructura: profe
Juan	1000	27	
Pedro	4000	40	
Carlos	120	1200	
Martín	2000	21	

Defina un arreglo de estructuras y asigne los datos anteriores, luego suba los sueldos de los profesores al doble, excepto a Carlos, quien debe ganar **20000** y corregir su edad a **25**. Reporte la tabla antes y después de la actualización. Ejemplo de salida:

Datos iniciales

(la tabla anterior)

Datos actualizados:

Nombre	Sueldo	Edad
Juan	2000	27
Pedro	8000	40
Carlos	20000	25
Martín	4000	21

Sugerencia: para identificar el caso especial, puede usar:

```

#include<string.h>
char carlos[] = "Carlos", dato[7];        // define variables
strncpy(dato, profe[i].nombre, 6)        // asigna valor a dato para la fila i
if(strcmp(carlos, dato)==0)                // identifica el caso especial

```