

Aula		Fila		Col.	
Nombre		1 ^{er} Ap.		2 ^o Ap.	
					Firma:

NOTAS DE OBLIGATORIA LECTURA

1. Las soluciones de examen entregadas escritas a lápiz, pueden ser anuladas
2. Sé ecológico: escribe por ambas caras de cada hoja de examen
3. Para la realización de este examen es obligatoria la presentación de identificación oficial actualizada con fotografía: carnet de estudiante, dni, pasaporte o permiso de conducir formato tarjeta
4. La parte obligatoria durará 2h 30'
5. No se admite el uso de variables globales en todo el examen

Parte obligatoria. A resolver por TODOS los alumnos

1.- (1 punto) Completa los huecos del siguiente código (desde Línea 1 hasta Línea 9), para que dicho programa sea capaz de controlar la carga de paquetes en un camión que puede llevar un máximo de 500 kgs. Este programa debe solicitar al usuario el peso del siguiente paquete a cargar, y clasificar dicho paquete como tipo A si su peso es menor de 20 kilos, de tipo B si está entre 20 y 49 kilos, de tipo C si está entre 50 y 100 kilos, y de tipo D si supera los 100 kilos. Sólo en el caso en el que el contenido actual del camión después de cargar dicho paquete no supere los 500 kilos, el programa solicitará otro paquete. El programa finaliza indicándole al usuario lo que debe pagar de acuerdo a los paquetes que ha cargado en el camión.

```

#include <stdio.h>
#define PRECIO_A 20
#define PRECIO_B 30
#define PRECIO_C 40
#define PRECIO_D 60
main ( ) {
    int nPesoTotal = 0, nPeso, pagar, a = 0, b = 0, c = 0, d = 0;
    do {
        printf("Peso del paquete: ");
        scanf("%d", &nPeso );

        if( _____ Línea 1 _____ )
            printf("Ese paquete ya no cabe en el camión.\n");
        else {
            if ( _____ Línea 2 _____ ) {
                printf("Paquete tipo A\n");
                _____ Línea 3 _____
            } else if ( _____ Línea 4 _____ ) {
                printf("Paquete tipo B\n");
                _____ Línea 5 _____
            } else if ( _____ Línea 6 _____ ) {
                printf("Paquete tipo C\n");
                _____ Línea 7 _____
            } else {
                printf("Paquete tipo D\n");
                _____ Línea 8 _____
            }
        }
        nPesoTotal = nPesoTotal + nPeso;
    } while ( _____ Línea 9 _____ );
    pagar = a * PRECIO_A + b * PRECIO_B + c * PRECIO_C + d * PRECIO_D;
    printf("Debe pagar %d\n", pagar );
}

```

2.- (1.5 punto) Implementar la función **veces** para calcular el número de veces que aparece un carácter en una cadena de caracteres.

Ejemplo de uso:

```
...
v = veces("Esto es una prueba de examen",'e');
printf("La e aparece %d veces.\n",v);
/* Escribiría: "La e aparece 5 veces ", que es el número de veces que aparece la letra 'e' */
....
```

3.- (4 puntos) Un “cuadrado casi mágico” es una matriz cuadrada de valores enteros en la que la suma de todos los elementos de cualquier fila, columna o la diagonal principal tiene como resultado el mismo valor:

8	1	6	=>	8 + 1 + 6 = 15	Diagonal principal = 8 + 5 + 2 = 15
3	5	7	=>	3 + 5 + 7 = 15	
4	9	2	=>	4 + 9 + 2 = 15	
15	15	15			

Se pretende realizar un programa en “C” que permita leer de fichero, visualizar y comprobar matrices tipo “cuadrado casi mágico” de cualquier dimensión. Para ello se propone que el programa pida al usuario el nombre del fichero de texto, lo cargue en la matriz y presente un menú con las siguientes opciones:

- 1.- Visualización por pantalla de un posible cuadrado casi mágico.
- 2.- Comprobación de un cuadrado casi mágico. Comprobará si la matriz cargada del fichero es un cuadrado mágico. En caso afirmativo, mostrará por pantalla “El cuadrado es casi magico”. En caso negativo mostrará por pantalla “El cuadrado NO es casi magico”.
- 3.- Salir del programa.

Implementa las cuatro funciones que aparecen en el programa principal siguiente:

```
#define N 100
main()
{
    int opcion, matriz[N][N];
    cargar(matriz);           // (1 punto)
    do {
        opcion = menu();     // (0.5 puntos)

        if ( opcion != 3 ) {
            switch ( opcion ) {
                case 1:
                    visualizacion(matriz);    // 1 punto
                    break;
                case 2:
                    comprobar(matriz);       // 1.5 puntos
                    break;
                case 3:
                    break;
            }
            system("pause");
        }
    } while ( opcion != 3 );
}
```