

Fundamentos de Programación I. Examen Ordinario. (24-01-19)

Nombre:..... Grupo..... DNI.....

1) (1.5 puntos) ¿Qué salida produciría el siguiente programa? Explicar por qué.

```
#include <stdio.h>
int a = 1;
int examen(int *b, int c);
void main(){
    int b = 2, c = 3;
    b = examen(&c, a);
    printf("Fuera: %d %d %d.\n",a, b, c);
}
int examen (int *b, int c){
    a = a + 1;
    c = c + 2;
    *b = *b + 4;
    printf("Dentro: %d %d %d.\n",a, *b, c);
    return c;
}
```

2) Queremos gestionar la compra de entradas de un cine, para ello vamos a representar el cine como una matriz de 20 filas y 40 butacas por fila, cada componente de la matriz contendrá 'L' o 'R' (libre o reservado).

a) (2 puntos) Implementa la función ***bool nLibres (char fila[], int n);*** que devuelve true si hay más de n butacas libres en el vector fila y false en caso contrario. (fila: es un vector que contiene 'L' o 'R' es una fila de la matriz cine, es decir, es una fila de la matriz cine)

b) (1.5 puntos) Realiza una función ***filasButacasLibre*** a la que le pasaremos como parámetros un cine (matriz) y un número entero *n*. La función nos devolverá un vector que contenga las filas en las que se encuentren *n* o más butacas libres juntas.

Utiliza la función implementada en el apartado anterior

bool nLibres (char fila[], int n);

Ejemplo:

CINE

L	L	O	O	L	L	L	O	O
L	O	O	O	L	O	L	L	O
O	O	O	O	O	L	L	O	O
L	L	L	O	O	L	L	L	L
L	L	O	L	L	L	L	O	O
L	O	O	O	L	O	L	L	O
L	L	L	L	L	L	L	O	O

Si n es 3 (al menos tres butacas libres juntas)

VECTOR DE SALIDA:

0	3	4	6	-1	-1	-1
---	---	---	---	----	----	----

- 3) (2.25 puntos) Dada una distribución de Poisson de media m , se define la probabilidad de que un suceso ocurra r veces como

$$Prob = \frac{m^r}{r!} e^{-m}$$

Crea un programa que lea por teclado los valores de m y r y calcule y muestre por pantalla la probabilidad de que el suceso ocurra como mucho r veces.

No puede usarse la función `pow` para calcular las potencias.

Recuerda que

$$e^{-m} = \frac{1}{e^m}$$

- 4) La DGT, de cara a cumplir el objetivo de reducción de víctimas en la carretera, y de paso conseguir unos eurillos extra, va a instalar unos nuevos radares de tramo, los cuales registrarán los instantes de paso en 2 puntos de control intermedios además del instante de entrada y de salida del tramo. Este nuevo funcionamiento hace necesario redefinir el sistema de gestión de los mismos.

Así, cuando un coche llega al punto de entrada del tramo, el sistema de ese punto abrirá un registro en el sistema en el que se anotarán, el identificador del tramo (código alfanumérico de 12 caracteres) la matrícula del coche, la fecha (día, mes y año) y la hora de paso (hora, minutos, segundos y décimas de segundo). Posteriormente, cuando el coche pasa por cada uno de los puntos de control, el sistema buscará la ficha correspondiente y en ella anotará la hora de paso con la misma precisión que en la entrada. Por último, cuando el coche llega al punto final del tramo se volverá a anotar el instante de paso y además se calculará y anotará en la ficha el estado del ese paso (CORRECTO, ADMISIBLE, SANCIONABLE).

Dado la limitación de recursos que tiene el sistema para cada tramo, se almacenarán los datos de como máximo 50 coches. Una vez alcanzado este límite, se transferirán los registros ya completados al servidor central de la DGT donde se completará el proceso.

Para implementar el nuevo sistema se pide

- a) (1.25 puntos) Definir todos los tipos de datos necesarios para almacenar la información en los sistemas de cada uno de los tramos.
- b) (1.5 puntos) Definir una función que dada la ficha correspondiente a un coche calcule el estado teniendo en cuenta la siguiente tabla.
 - CORRECTO. El tiempo de cada uno de los tramos no es superior a 3 minutos
 - ADMISIBLE. Hay como máximo un tramo en el que se ha superado los 3 minutos, pero el tiempo total no excede de 15 minutos
 - SANCIONABLE. El tiempo total excede de 15 minutos.

Fundamentos de Programación I.

Ordinario. Prácticas. (24-01-19)

Nombre:..... Grupo..... DNI.....

Dada la siguiente estructura de datos, utilizada para representar un polinomio,

```
#define N 15
typedef struct {
    int exp;
    float coef;
}termino;

typedef struct _polinom {
    int numTerminos;
    termino t[N];
}polinomio;
```

Los términos no tienen que estar en orden del exponente. El polinomio no tendrá más de un termino con el mismo exponente (ej = $3x^2 + 5x^3 + x$)

1. Realizar las siguientes funciones para el tratamiento de polinomios:

- a) //introduce *numTerm* de monimios (terminos)en el polinomio *pol*
void **introducePol**(polinomio *pol, int numTerm) ;
- b) //Sumará dos polinomios *pol1* y *pol2* y devuelve la suma
void **sumapolinomio**(polinomio pol1, polinomio pol2,polinomio *suma);
- c) //muestre el polinomio por pantalla en su formato tradicional
(ej $3x^2 + 5x^3 + x$.)
void **imprime**(polinomio pol);
- d) // devuelve el grado de un polinomio
int **grado** (polinomio p);
- e) //modifica el polinomio, cambiando el signo de todos los términos del polinomio
void **cambiaSigno** (polinomio *p);
- f) Completa el programa principal para poder probar las funciones.
(completar)

NOTAS PARA LA ENTREGA.

- Debes entregar esta hoja con tu nombre y apellidos.
- En la tarea habilitada en moodle sube el fichero .c donde has escrito el programa. No olvides poner al principio del fichero tu nombre, apellidos, grupo.