

## Examen de Metodología y Tecnología de la Programación. (Junio 03).

1. En un observatorio meteorológico se tienen guardados en un fichero de tipo texto denominado "Datos.txt" los valores de lluvia mensuales de un periodo de tiempo. En el primer registro del fichero están almacenados dos números enteros que se corresponden con el número de años de la serie y con el año inicial y a continuación están guardados los datos mensuales en formato float, de forma tal que cada año ocupa una línea en el fichero, tal como se refleja debajo:

```
25    1975
23.5  23.6  45.9  34.9  25.9  34.7  34.8  12.8  89.5  12.8  12.6  5.3
34.6  45.6  56.1  32.1  1.2   23.6  7.8   7.9   15.4  16.5  12.5  17.3
```

.....  
.....

Implementar un programa en C que realice de manera secuencial (sin usar menú) las siguientes acciones:

- Lectura de los datos de lluvia del fichero y almacenamiento de los mismos en una matriz dinámica.
- Almacenamiento en un vector de los datos medios mensuales de lluvia para cada uno de los meses del año.
- Almacenamiento en un vector de los datos anuales de lluvia para cada uno de los años de la serie.
- Obtención del mes y del año en el cual se haya producido la precipitación máxima y la mínima.
- Almacenamiento de los datos de lluvia mensuales correspondientes a los años cuya precipitación anual sea inferior a 300 mm. en un fichero binario denominado "sequia.bin".

Cada uno de los apartados se implementará en una función, y todas han de ser de tipo void.

2. En una estructura tipo lista se quiere almacenar las estaturas en cm. de una muestra de personas de una población. En cada elemento de la lista se almacenará una estatura y el número de veces que ésta se repite en la muestra de personas (frecuencia). Implementar un programa en C que posea un menú con las siguientes opciones:
- Dada una estatura, insertarla en orden creciente en la lista.
  - Dada una estatura, borrar un elemento de la lista con esa estatura teniendo cuenta la frecuencia de ésta.
  - Listar en orden decreciente las estaturas almacenadas en la lista con sus correspondientes frecuencias.
  - Cálculo de la estatura media de los elementos almacenados en la lista.
  - Borrado completo de la lista.

Cada uno de los apartados se implementará en un función.

## METODOLOGIA DE LA PROGRAMACION SEPTIEMBRE 2003

### PROBLEMAS

1.- Una lista simplemente enlazada está formada por un conjunto de elementos. Cada elemento contiene un entero y una cadena de 30 caracteres. Se supone que la lista está creada y que en ella están insertados un número desconocido de elementos, siendo **inicio** el identificador del puntero que apunta al primer elemento de la lista.

- Codificar una función que indique cuantos elementos hay en la lista.
- Codificar una función que elimine de la lista aquellos elementos cuyo valor entero sea par y mayor de 50.
- Codificar el procedimiento mas simple posible que visualice los elementos de la lista como si la misma fuera una "pila".

2.- Se desean almacenar en una matriz dinámica los datos correspondientes a las calificaciones obtenidas en una serie de asignaturas por un conjunto de alumnos.

Se supone que todos los alumnos han cursado una titulación de 6 cursos y que en cada curso hay 5 asignaturas, todas con idéntico número de créditos. En cada fila deben aparecer las 5 calificaciones de un alumno, en cada columna la calificación correspondiente a una asignatura y una tercera dimensión las calificaciones correspondientes a cada curso, tal y como se muestra en el ejemplo:

En un fichero denominado alumnos.bin se encuentran almacenados los nombres de los alumnos junto con un identificador (número entero), según la plantilla struct alumno { char nombre [100]; int n;};

- Codificar una función que calcule el numero de estudiantes que hay en el fichero, cree la matriz necesaria e inserte las calificaciones de cada estudiante en la fila de la matriz correspondiente al identificador del alumno en el fichero.
- Codificar una función que reciba el nombre de un alumno y devuelva en un vector las calificaciones medias del estudiante en cada uno de los 6 cursos.
- Obtener el nombre del estudiante que posee mejor expediente académico (media de las medias de cada uno de los 6 cursos).

## Metodología de la Programación. Septiembre 2003.

Nombre y Apellidos:

Especialidad:

### CUESTIONES:

1.- Se tiene definida una variable *int x*, global, y una variable *int x*, local en el interior del *main*. ¿Cuál tiene prioridad?

- La global
- La local
- Dentro del *main* la local y fuera del *main* la global

2.- Indica cual de las siguientes afirmaciones sobre los métodos de ordenación es correcta.

- El método quicksort es el más rápido de todos, independientemente del número de elementos a ordenar.
- El método de selección tiene un mejor comportamiento que el quicksort cuando el número de elementos a ordenar es relativamente pequeño.
- El método burbuja es el mejor de los métodos de orden  $n^2$ .

3.- Se tienen las siguientes declaraciones:

```
int pepe(void);
```

```
void juan(int);
```

¿Es correcta la expresión *juan(pepe())* escrita dentro de *main()*?

- No, si se quiere pasar el valor que devuelve *pepe* a *juan* es necesario escribir *x=pepe(); juan(x);*
- Si
- Si, pero la expresión no debe estar en el *main()* sino en la zona de declaraciones globales.

4.- Se tienen las siguientes líneas de código. El propósito de las mismas es recoger en la dirección del campo *x* de la estructura, la dirección del campo *y* de la estructura.

```
struct dato {int x; int y;};
```

```
struct dato x, *y;
```

```
y = &x;
```

```
&(x.x)=y->y;
```

¿Cuál es correcta entre las respuestas?

- Todo es correcto.
- Hay dos errores. No puede ser que haya una variable "x" de estructura cuando hay un campo llamado "x" en la estructura, y el valor del campo "y" es lo que se coloca en la dirección del campo "x".
- Hay un error, ya que no se puede modificar la dirección del campo *x*.

5.- ¿Cual de las siguientes afirmaciones es correcta con relación al siguiente esquema?

```
for (i=0;i<i+3)
{
    i=i-3;
}
```

- El esquema nunca finaliza, es infinito.
- No se entra en el esquema.
- Se entra la primera vez y se abandona tras la primera iteración.

6.- La instrucción "break" dentro de un *switch*

- Sirve para abandonar el esquema *switch* en cuanto se encuentra.
- Es obligatorio su uso dentro de cada "case".
- Si no se utiliza el esquema *switch* funciona como un esquema iterativo.

7.- El anidamiento de esquemas condicionales del tipo *if-else* dentro de un programa

- Tiene como límite máximo 50 anidaciones dentro del mismo esquema.
- No tiene ningún límite en cuanto al número de anidaciones.
- Existe un límite en el número de anidaciones que depende del compilador que se utilice, aunque este límite es tan alto que en la práctica no hay que preocuparse.

8.- Si *x* es una variable de tipo *int* y *p* es una variable de tipo *int \**, la instrucción  $\&(\&x)=p$ ;

- No es correcta porque debe ser  $p=\&(\&x)$ ;
- No es correcta.
- Es correcta.

9.- Una función debe recibir un parámetro que es un número real, para utilizarlo sin modificarlo, y un vector de enteros, para modificarlo. ¿Cual es el prototipo correcto?

- void funcion(float, int \*)*;
- void funcion(double, int \*)*;
- a) y b) son correctas.

10.- El algoritmo

*Algoritmo p(n)*

*Inicio*

*Si n<1*

*X<- 1*

*Y<- 2*

*Devolver (x\*y)*

*Sino*

*X<- 1*

*Y<- 2*

*Devolver p(n-1)*

*Finsi*

*Fin*

Tiene como tiempo de ejecución

- $2(n+1)$
- $2n^2$
- $n + 2$

Contenido del fichero DATOS . TXT

5 1992

19	92	23.32	12.45	16.76	19.83	34.40	23.08	54.78	32.00	34.12	56.32	54.45
23.23	33.12	42.67	76.77	56.63	43.49	65.09	12.89	43.87	54.87	74.65	48.66	
70.57	80.66	40.80	91.98	60.75	47.34	23.23	22.43	66.54	58.11	99.10	86.01	
14.23	13.23	16.43	18.55	18.44	13.33	12.22	16.22	17.54	16.34	18.67	18.77	
36.60	54.98	75.99	55.88	40.76	87.23	22.32	43.12	43.22	43.45	54.23	42.12	