



Universidad
de Huelva

Departamento de Ingeniería Electrónica,
Sistemas Informáticos y Automática.
Área de Ciencias de la Computación e Inteligencia Artificial

Metodología de la Programación - Febrero 2002 - Parte I

Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)

Sección informativa :

- El examen se compone de tres ejercicios divididos en 2 partes.
- La presente es la Parte I, tiene una duración de 1 h. y 15 m. para realizarlo.
- Solo se podrán realizar consultas durante los primeros 20 minutos.
- Se agotará la convocatoria transcurridos los primeros 20 minutos.
- Tras un descanso de 15 minutos comenzará la Parte II.

Ejercicio 1: (2,5 Puntos) Contestar en los propios folios del enunciado.

En el siguiente programa escrito en ANSI C faltan varias cosas, las cuales vienen indicadas con ____.
También puede haber lugares donde se indique ____ y no sea necesario escribir nada. Lo único que no tiene que hacer es tachar cosas ya escritas. El programa toma desde teclado el tamaño de una tabla, la crea, la rellena aleatoriamente, la visualiza y finalmente la ordena con el clásico método de la burbuja.

```
#include <stdlib.h>

_____rellenar(int __A,int num)
// devuelve tanto por parámetro como por la propia llamada a función,
// un puntero a la tabla A rellena de forma aleatoria con n
// elementos.
{
    int i;
    for (i=0;i<num;i++)
        A[i]=random(1000);

    return A;
}

void visualizar(int __A,int n)
{
    int i;
    for (i=0;i<n;i++)
        printf ("%d ",__A[i]);
}

void burbuja (int __A, int n)
{
    int i,j;

    for (i=0;i<=n-2;i++) {
        for (j=n-1;j>=i+1;j--)
            if (A[j-1] >A[j])
                intercambio(____,____); // intercambiar A[j-1] y A[j]
    }
}

void intercambio(int __a, int __b)
{
    int temp;

    ____temp=__a;
    ____a=__b;
}
```

```
    __b=__temp;
}
void main(void)
{
    int cantidad;
    int *Tabla;

    randomize(); // inicializa generador números aleatorios
    printf("Teclear el numero de elementos");
    scanf(____cantidad);
    Tabla= _____; //reserva de memoria
    rellenar (Tabla, cantidad);
    visualizar(Tabla, cantidad);
    burbuja(Tabla, cantidad);
    visualizar (__Tabla,cantidad);
    burbuja (__Tabla,cantidad);
}
```

Ejercicio 2: (3,5 Puntos en total) Contestar en los otros folios del enunciado. No olvidar nombre.

a) Defina brevemente (3 ó 4 líneas) los conceptos de Programación Orientada a Objetos que se proponen a continuación. **(0,75 Puntos)**

- Polimorfismo.
- Herencia.
- Clase derivada.
- Método virtual.
- Puntero a objeto.

b) Busca los errores que existen en el código C++ siguiente y rescríbelo correctamente, razonando los cambios. **(1 Punto)**

```
class base{
    private:
        int p;
    public:
        void base(int cuanto){p=cuanto;}
        void ~base (void){}
};

class derivada: public base{
    protected:
        int p2;
    public:
        void derivada(int cuanto){
            p=cuanto;
            p2=p+1;
        }
        void ~derivada(void){}
};

void main (void){
    base unaclase=new derivada(5);
    unaclase.p2=unaclase.p2+1
};
```

c) Diseña una clase que permita implementar una cola de enteros. Para ello, uno de los atributos será una tabla dinámica de enteros. Dentro de conjunto de métodos a implementar deben figurar los siguientes: **(1,75 Puntos)**

```
void insertar(int el);
int extraer (void);
int colavacia (void);
void visualizar(void), /* Sin extraer los elementos */
```



Metodología de la Programación - Febrero 2002 - Parte II

*Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)*

Sección informativa :

- La presente es la Parte II, tiene una duración de 1 h. y 15 m.
- Solo se podrán realizar consultas durante los primeros 15 minutos.

Ejercicio 3: (4 Puntos en total)

Escriba lo que falta al programa dado en ANSI C. El programa toma dos archivos de texto en disco, (con el formato indicado en el ejemplo de más abajo) de tamaño y número de elementos desconocidos a priori, y cuyos elementos no siguen ningún orden conocido, y graba un tercer archivo de texto, que contiene todos los elementos ordenados de menor a mayor y su número de apariciones (según formato indicado más abajo).

Ejemplo de Formato de los archivos de entrada.

Primer archivo: 123.43 9034.775 34 234.1 2335 3.75
Segundo archivo: 234.1 34 9598.4 234.1

Ejemplo de Formato del archivo de salida:

3.75 1
34 2
123.43 1
234.1 3
2335 1
9034.775 1
9598.4 1

Necesariamente debe:

- Escribir código que controle los posibles errores de apertura de los archivos.
- Tener en cuenta que un mismo número puede aparecer varias veces tanto en un mismo archivo como en ambos.
- Tomar los nombres de los 3 archivos por teclado.

```
#include <stdlib.h>
#include <stdio.h>

struct tab frecuencia {
    double numero;
    int repeticion;
};

// lee fichero e inserta en tabla:
double *cargar (double *A,int *n);

// crea la tabla de frecuencias partiendo de una y otra tabla:
struct tab_frecuencia* crear_nueva_tabla(double *Tab_A,int a,double *Tab_B,int b,
struct tab_frecuencia *Tab_C,int *c);

// inserta por orden en la nueva tabla:
struct tab_frecuencia *insertar(struct tab_frecuencia *Tab_C, int *n,double valor);

// graba la nueva tabla en disco con el formato del enunciado:
void grabar(struct tab_frecuencia *A,int n);

int main(void)
{
    double *TA,*TB;
    struct tab_frecuencia *TC;
    int n_A,n_B,n_C=0;

    TA=TB=NULL;
    TA=cargar(TA,&n_A);
    if (TA!=NULL)
    {
        TB=cargar(TB,&n_B);
        if (TA!=NULL)
        {
            TC=crear_nueva_tabla(TA,n_A,TB,n_B,TC,&n_C);
            grabar(TC,n_C);
        }
        else
            printf("Falta el segundo fichero\n");
    }
    else
        printf("Falta el primer fichero\n");

    return 0;
}
```

```

double *cargar (double *A,int *n)
// lee un fichero e inserta los datos en la tabla A
{
    Escriba el código que corresponda (1 punto)
}

struct tab_frecuencia *crear_nueva_tabla(double *Tab_A,int a, double *Tab_B,int b,
struct tab_frecuencia *Tab_C,int *c)
// Crea la nueva tabla en función de las originales, Tab_A y Tab_B.
// "a" y "b" son el número de elementos de dichas tablas.
// La de salida es Tab_C, y "c" es el número de elementos de Tab_C.
// Utiliza la llamada a la función "insertar" para hacerlo en orden.
{
    Escriba el código que corresponda (0.75 puntos)
}

struct tab_frecuencia *insertar(struct tab_frecuencia *Tab_C, int *n,double valor)
// Toma la tabla, su tamaño y un valor a insertar, y lo inserta
// por orden en su lugar para que quede ordenada de menor a mayor.
// Devuelve la misma tabla con el elemento insertado y
// el número de elemencos.

// En caso de que el elemento ya se encontrase allí, solo aumenta su
// frecuencia de aparición.
{
    Escriba el código que corresponda (1.5 puntos)
}

void grabar(struct tab_frecuencia *A,int n)
// graba en disco la tabla A de "n" elementos
{
    Escriba el código que corresponda (0.75 puntos)
}

```