



EXAMEN DE SISTEMAS INFORMÁTICOS INDUSTRIALES (SOLUCIÓN) (TEORÍA)

La duración del examen es de 2 horas.

JUNIO 2016

1. Indicar si las siguientes afirmaciones son verdaderas o falsas:

- (a) En C++ para cada clase se deben implementar 2 destructores: uno que se llama cuando se destruye un objeto de la clase, y otro que se llama cuando se destruye un puntero a un objeto de la clase. F
- (b) Para implementar una arquitectura cliente/servidor en C ó C++ sólo puede utilizarse la librería de programación de sockets. F
- (c) El algoritmo de búsqueda binaria permite buscar un elemento dentro de un vector siempre y cuando el vector esté ordenado. V
- (d) La sentencia en C: `int vector[]={1,2,3};` es correcta a pesar de que no se haya indicado el tamaño del vector entre corchetes. V
- (e) El paso por referencia de un parámetro a una función en C++ permite que se modifique en la función el valor de la variable original pasada, sin necesidad de emplear notación de punteros. V
- (f) Las variables miembro de una clase base en C++ declaradas como privadas pueden ser accedidas desde una clase derivada de la clase base. F
- (g) En C++ se puede definir el operador + como un método de una clase o como una función global. V
- (h) La función `scanf` no puede utilizarse para leer una cadena de caracteres por teclado si ésta contiene espacios en blanco. V

(2 puntos)

2. Se desea crear una clase en C++ denominada **CTexto** que permita trabajar con cadenas de caracteres almacenadas en una matriz dinámica de caracteres. La clase tendrá las siguientes variables miembro privadas:

- `int líneas, columnas;` // número de líneas y columnas de la matriz
// de caracteres
- `int numLineas;` // número de líneas de texto (frases) introducidas
// en la matriz
- `char **ptexto;` // matriz de caracteres

Se pide:

- (a) Implementar el constructor por defecto que inicializa `ptexto` a nulo, y `líneas, columnas` y `numLineas` a 0. (0.25 puntos)
- (b) Implementar el constructor que recibe dos parámetros (número de líneas y número de columnas) y reserva memoria de forma dinámica para una matriz de caracteres según las líneas y columnas indicadas, inicializando el resto de variables enteras de forma oportuna. (0.75 puntos)

- (c) Implementar el destructor. (0.5 puntos)



Escuela Politécnica Superior de Elche

Grado en Ingeniería Electrónica y Automática Industrial

- (d) Implementar el método void `AñadirLineaTexto` (`char *cadena`) que permite añadir la cadena de caracteres que recibe como parámetro a una nueva línea de la matriz `ptexto`. En el caso de que no quede espacio para almacenar una nueva línea o no se haya reservado memoria para la matriz, no podrá almacenarse la cadena, mostrándose un mensaje de error.

(1 puntos)

Notas:

- No podrá hacerse uso de la función `strcpy`
- Para simplificar se considerará que el número de caracteres (incluyendo el `'\0'`) de cadena es menor o igual que el número de columnas de `ptexto`.

Ejemplo:

```
CTexto ejemplo(3,250);
ejemplo.AñadirLineaTexto("Esto es una cadena de
ejemplo");
ejemplo.AñadirLineaTexto("Esto es otra");
ejemplo.AñadirLineaTexto("Esto es una tercera
cadena");
```

Tras la ejecución de los 3 métodos, el contenido de `ptexto` sería:

```
Esto es una cadena de ejemplo
Esto es otra
Esto es una tercera cadena
```

- (e) Implementar el método void `MostrarTexto` (`void`) que muestre por pantalla todas las líneas de la matriz `ptexto`.

(0.5 puntos)

Solución:

(a)

```
CTexto::CTexto(void)
{
    ptexto=NULL;
    lineas=0;
    columnas=0;
    numLineas=0;
}
```

(b)

```
CTexto::CTexto(int nlin, int ncol)
{
    lineas=nlin;
    columnas=ncol;
    ptexto=new char*[lineas];
    for(int i=0;i<lineas;i++) ptexto[i]=new char[columnas];
    numLineas=0;
```



(c)

```
CTexto::~~CTexto(void)
{
    for(int i=0;i<lineas;i++) delete []ptexto[i];
    delete []ptexto;
}
```

(d)

```
void CTexto::AñadirLineaTexto (char *cadena)
{
    if(ptexto==NULL){
        cout<<"No se puede añadir la cadena\n";
    }
    else{
        if(numLineas==lineas){
            cout<<"No se puede añadir la cadena\n";
        }
        else{
            int i;
            for(i=0;cadena[i]!='\0';i++){
                ptexto[numLineas][i]=cadena[i];
            }
            ptexto[numLineas][i]='\0';
            numLineas++;
        }
    }
}
```

(e)

```
void CTexto::MostrarTexto (void)
{
    for(int i=0;i<numLineas;i++){
        printf("%s\n", ptexto[i]);
    }
}
```

3. Crear un programa en C que muestre las palabras de una cadena que coincidan con la longitud indicada.

(2.5 puntos)

- a. El programa deberá solicitar una cadena con un máximo de 100 caracteres (formada por sólo letras, las palabras separadas por espacios y sin punto al final), y un número entre 1 y 10. Se asumirá que tanto la cadena como el número introducido son correctos no siendo necesario realizar ninguna comprobación.



Escuela Politécnica Superior de Elche

Grado en Ingeniería Electrónica y Automática Industrial

- El programa deberá mostrar únicamente las palabras de la cadena que coincidan con la longitud indicada.
- Se muestran a continuación 3 ejemplos de funcionamiento de la ejecución del programa solicitado:

Introduzca una cadena: La casa del perro es de color azul verdoso y además el gato quiere usar su casa Introduzca un numero entero entre 1 y 10: 5 perro color	Introduzca una cadena: La casa del perro es de color azul verdoso y además el gato quiere usar su casa Introduzca un numero entero entre 1 y 10: 4 casa azul gato usar casa	Introduzca una cadena: La casa del perro es de color azul verdoso y además el gato quiere usar su casa Introduzca un numero entero entre 1 y 10: 2 La es de el su
---	---	---

Solución:

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

int main(int argc, char *argv[])
{
    char cad[100];
    int num,len,i,j,h,cont;

    // Solicitud de cadena y numero
    printf("Introduzca una cadena: ");
    gets(cad);
    printf("Introduzca un numero entero entre 1 y 10: ");
    scanf("%d",&num);

    len=strlen(cad); // Calculo de la longitud de la cadena introducida

    for(i=0;i<len;) // Bucle para recorrer la cadena
    {
        for(cont=0,j=i;j<len;j++,cont++) // Se busca la primera palabra
        {
            if(cad[j]==' ') // Cuando se encuentre el siguiente
                break; // espacio se finaliza el bucle
        }
        if(cont==num) // Ahora, si la longitud de la palabra coincide
        { // con la solicitada se muestra
            for(h=i;h<j;h++) // La palabra comienza en la
            { // posición "i" y finaliza en la posición "j"
                printf("%c",cad[h]);
            }
            printf("\n"); // Se muestra un salto de linea
            // para separar las palabras
        }
        i=j+1; // Se actualiza el contador "i" para comenzar
            // al inicio de la siguiente palabra, +1 para evitar el espacio
    }
    system("PAUSE");
}
```



Escuela Politécnica Superior de Elche

Grado en Ingeniería Electrónica y Automática Industrial

4. Crear un programa en C que lea una matriz cuadrada de un fichero de texto y realice los cálculos necesarios para saber si se trata de un cuadrado mágico.

(2.5 puntos)

- El fichero que contiene la matriz tiene en la primera fila un número que indica la dimensión de la matriz (sólo un valor puesto que es cuadrada). La matriz está formada por números enteros.
- Para que una matriz sea considerada un cuadrado mágico será necesario que la suma de cada una de sus filas así como de cada una de sus columnas de el mismo valor.
- Aspectos para resolver el problema:
 - El nombre del fichero se pasará por línea de comandos.
 - La matriz leída debe almacenarse en memoria. Tendrá una dimensión máxima de 5x5 por lo que no será necesario realizar una reserva dinámica de memoria para almacenarla.
 - Se deberá reservar dinámicamente memoria para almacenar el resultado de la suma de cada fila y columna. Se calcularán dichas sumas y se almacenarán en el vector.
 - Finalmente se comprobará si se trata de un cuadrado mágico y se indicará por pantalla.
- Se presentan a continuación 4 ejemplos de funcionamiento con 4 ficheros diferentes:

Contenido del fichero matriz1.txt 3 8 1 6 3 5 7 4 9 2	Contenido del fichero matriz2.txt 3 8 2 7 3 4 6 5 9 1
examen.exe matriz1.txt Suma de filas: 15 15 15 Suma de columnas: 15 15 15 SI es un cuadrado magico	examen.exe matriz2.txt Suma de filas: 17 13 15 Suma de columnas: 16 15 14 NO es un cuadrado magico
Contenido del fichero matriz3.txt 2 5 3 7 2	Contenido del fichero matriz4.txt 4 4 5 16 9 14 11 2 7 1 8 13 12 15 10 3 6
examen.exe matriz3.txt Suma de filas: 8 9 Suma de columnas: 12 5 NO es un cuadrado magico	examen.exe matriz4.txt Suma de filas: 34 34 34 34 Suma de columnas: 34 34 34 34 SI es un cuadrado magico

Solución:

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    char nombre_fichero[50];
    FILE *fich;
    int dim,i,j,a=0,iguales=1;;
    int mat[5][5];
```



Escuela Politécnica Superior de Elche

Grado en Ingeniería Electrónica y Automática Industrial

```
int *c;

        // Si no se introducen un segundo argumento
if(argc<2) { // como nombre de fichero se indica y se termina
printf("No se ha introducido el nombre del fichero\n");
return -1;
}

        // Se abre el fichero en modo lectura y si no
fich=fopen(argv[1], "r"); // se abre correctamente se indica y se termina
if (fich== NULL) {
printf("Error abriendo el fichero\n");
return -1;
}

fscanf(fich,"%d",&dim); // Se lee el primer valor que
//indica la dimension de la matriz cuadrada
for (i=0;i<dim;i++) { // Se lee con un doble bucle cada valor
for (j=0;j<dim;j++) {
fscanf(fich,"%d",&mat[i][j]);
}
}

c = (int*)malloc(2*dim*sizeof(int)); // Se reserva memoria para
// almacenar la suma de filas y columnas
for (i=0;i<dim;i++) {
c[a]=0;
c[a+dim]=0;
for (j=0;j<dim;j++){
c[a]+mat[i][j]; // Se suman los valores de cada fila
c[a+dim]+mat[j][i]; // Se suman los valores de cada columna
}
a++;
}
// Se muestran los resultados
printf("Suma de filas: ");
for (i=0;i<dim;i++)
printf("%d ",c[i]);
printf("\nSuma de columnas: ");
for (i=0;i<dim;i++)
printf("%d ",c[i+dim]);

// Comprobacion de si todos los valores son iguales
for(i=0;i<2*dim-1;i++) {
if(c[i]!=c[i+1])
iguales=0;
}
// Si todos son iguales se indica que se trata de un cuadrado magico
if(iguales==1)
printf("\nSI es un cuadrado magico\n");
else
printf("\nNO es un cuadrado magico\n");
}
```