



EXAMEN DE SISTEMAS INFORMÁTICOS INDUSTRIALES (SOLUCIÓN) (TEORÍA)

JUNIO 2015

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

- (a) En C++, cuando se destruye un objeto de una clase derivada, se ejecuta primero el destructor de la clase base y después el de la clase derivada. **F**
- (b) Para implementar una arquitectura cliente/servidor en C ó C++ sólo puede utilizarse la librería de programación de sockets. **F**
- (c) Una clase abstracta es una clase que tiene al menos una función virtual pura. **V**
- (d) En C, cuando se pasa un vector como parámetro a una función, se pasa siempre por valor. **F**
- (e) En C++ no se puede sobrecargar el operador << para poder mostrar el contenido de un objeto con *cout*. **F**
- (f) C++ permite dar valores por defecto a los parámetros de una función. **V**
- (g) El identificador de un array es en realidad un puntero al elemento 0 del array. **V**
- (h) En C es posible leer un fichero carácter a carácter. **V**

(2 puntos)

2. Crear un programa que lea la información de una matriz introducida mediante la línea de comandos y realice ciertas operaciones. El primer parámetro corresponderá al número de filas de la matriz y el segundo al número de columnas. El resto de valores corresponderán a los valores de la matriz (serán valores enteros).

- (a) Se deberá comprobar que el número de parámetros de entrada coincide en función del número de filas y columnas introducidas. (0.25 puntos)
- (b) Se realizará una reserva dinámica de memoria para la matriz de entrada. (0.75 puntos)
- (c) Se almacenarán los valores introducidos en la línea de comandos en la matriz. (0.75 puntos)
- (d) Mostrar por pantalla el mínimo, máximo y media de los valores de la matriz. (0.75 puntos)

NOTA: se puede hacer uso de la función *atoi* que convierte una cadena a un entero: `int atoi(const char *str)`.

Ejemplo de ejecución:

```
ejercicio.exe 2 3 -6 4 8 -3 6 -4
```



Escuela Politécnica Superior de Elche

Grado en Ingeniería Electrónica y Automática Industrial

Contenido del fichero.txt generado:

```
Minimo: -6
Maximo: 8
Media: 0.83
```

Solución:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i,j,h, filas,columnas, minimo,maximo;
    float media=0;
    int **matriz;
    // Comprobación de parámetros de entrada
    if (argc==1){
        printf("No ha introducido parametros. ejercicio.exe
filas columnas valores ...\n");
        return -1;
    }
    filas=atoi(argv[1]);
    columnas=atoi(argv[2]);

    if ((argc-3)!=(filas*columnas)){
        printf("El numero de parametros introducidos no es
correcto\n");
        return -1;
    } else if (filas==0 || columnas==0){
        printf("El numero de filas y columnas no puede ser 0.");
    }
    // Reserva dinámica de memoria
    matriz = (int **)malloc(filas*sizeof(int*));
    for(i=0; i < columnas; i++)
        matriz[i] = (int *)malloc(columnas*sizeof(int));
    // Leer valores y almacenar en matriz
    // Simultaneamente se comprueba el mínimo y máximo, y la
media
    h=3;
    minimo=atoi(argv[h]);
    maximo=minimo;
    for (i=0;i<filas;i++){
        for (j=0;j<columnas;j++){
            matriz[i][j]=atoi(argv[h]);
            if(matriz[i][j]>maximo) maximo=matriz[i][j];
            if(matriz[i][j]<minimo) minimo=matriz[i][j];
            media+=matriz[i][j];
            h++;
        }
    }
    media/=filas*columnas;
    // Mostrar datos por pantalla
    printf("Minimo:          %d\nMaximo:          %d\nMedia:
%.2f\n",minimo,maximo,media);

    // Liberar memoria (aunque no se solicitó en el examen es
necesario que recordéis que es importante liberar la memoria
reservada cuando finalice un programa)
```



```
for (i=0; i<filas; i++)
    free(matriz[i]);
free(matriz);
}
```

3. Se desea crear una clase en C++ denominada **CComando** que permita definir comandos matemáticos. La clase tendrá las siguientes variables miembro privadas:

- `int n;` // número de operandos
- `float *pvalores;` // vector dinámico unidimensional para almacenar // los operandos
- `char operador;` // sólo admitirá los valores '+' y '*'

Por ejemplo, si se quiere definir el comando "4+3+6+2" (se considera que no es posible mezclar operadores dentro de un mismo comando), se crearía un objeto con las siguientes variables miembro:

- `n=4`
- `pvalores={4,3,6,2}`
- `operador='+'`

Se pide:

- (a) Implementar el constructor por defecto que define el comando "0+0". (0.5 puntos)
- (b) Implementar el constructor que recibe tres parámetros (número de operandos, vector con los valores de los operandos, carácter que define el operador). Ejemplo:

```
int v[4]={4,3,6,2};
CComando sumar(4,v,'+');
```

(0.75 puntos)
- (c) Implementar el destructor. (0.5 puntos)
- (d) Implementar el método `Ejecutar` que debe devolver el resultado de ejecutar el comando (tanto cuando el operador es '+' como cuando el operador es '*'). Por ejemplo, si se ejecutara `sumar.Ejecutar()` considerando el objeto `sumar` creado anteriormente, se obtendría el valor de la suma de todos los operandos, es decir, 15. (1.25 punto)

Solución:

(a)

```
CComando::CComando(void)
{
    n=2;
    pvalores=new float[n];
    pvalores[0]=0;
    pvalores[1]=0;
    operador='+';
}
```



(b)

```
CComando::CComando(int num, float *v, char op)
{
    n=num;
    pvalores=new float[n];
    for(int i=0; i<n; i++) pvalores[i]=v[i];
    operador=op;
}
```

(c)

```
CComando::~~CComando(void)
{
    delete []pvalores;
}
```

(d)

```
float CComando::Ejecutar(void)
{
    int i;
    float res;
    if(operador=='+'){
        res=0;
        for(i=0;i<n;i++) res+=pvalores[i];
    }
    else{
        res=1;
        for(i=0;i<n;i++) res*=pvalores[i];
    }
    return(res);
}
```

4. Escribir un programa que pida al usuario que introduzca una frase por teclado (para almacenar la cadena de caracteres utilizar un array de caracteres de tamaño 100). La frase estará compuesta por un máximo de 15 palabras, y se considera que las palabras están separadas únicamente por un espacio (no aparecerán otros símbolos como ',', ';', '-', '.', ...). El programa debe mostrar por pantalla las palabras de la frase en orden inverso, es decir, desde la última palabra hasta la primera. No es necesario hacer uso de reserva dinámica de memoria. Ejemplo de ejecución (en negrita aparece la cadena introducida por el usuario):

(2.5 puntos)

| |
|-----------------------------------------------------------------------------------------------------------------|
| Introduce frase: La pluma es la lengua de la mente Frase invertida: mente la de lengua la es pluma La |
|-----------------------------------------------------------------------------------------------------------------|

Solución:



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

void main (void)
{
    char cadena[100];
    char palabras[15][100]
    int i,j,numPal=0;

    printf("Introduce cadena:");
    gets(cadena);

    for(i=0;cadena[i]!='\0';i++){
        j=0;
        while(cadena[i]!=' ' && cadena[i]!='\0'){
            palabras[numPal][j++]=cadena[i++];
        }
        palabras[numPal][j]='\0';
        numPal++;
        if(cadena[i]=='\0') break;
    }

    for(j=numPal-1;j>=0;j--){
        printf("%s ", palabras[j]);
    }
}
```