

FUNDAMENTOS DE PROGRAMACIÓN. 1º ING. INFORMÁTICA (Plan 2000)
1ª Convocatoria. 21 de Enero de 2002

NORMAS DE EXAMEN: Notas inferiores a 3.5 en este ejercicio implican suspender la asignatura (en primera convocatoria). Las notas superiores a 3.5 promediarán con la calificación del proyecto de laboratorio según la siguiente proporción: 60% nota del examen, 40% nota del proyecto.

Tiempo: Para realizar este examen se dispone de 2:30 de tiempo

1. Escribe una función que lea la cantidad de líneas y letras del abecedario de un fichero de texto (mayúsculas y minúsculas). No se consideraran ni espacios en blanco, ni signos de puntuación ni caracteres especiales. Guardar esta información en otro fichero texto siguiendo el siguiente formato: (2 puntos)

```
Líneas : 10
Letras : 54
```

El prototipo de la función debe ser el siguiente, siendo *nombrefichero* el nombre del archivo que se desea analizar y el valor de retorno de la función indicará si se ha producido (o no) algún error al manipular los archivos.
(2 puntos)

```
bool ProcesarFichero(string nombrefichero);
```

2. Haz una función a la que se le pase como parámetro una cadena que almacena la fecha en formato DD/MM/AA y la convierta al formato Día de Mes de Año. Por ejemplo la fecha 01/03/01 se debe convertir a la cadena 1 de 3 de 2001.
(2 puntos)
3. Escribe una función recursiva que devuelva como resultado la longitud de un string sin utilizar la función de string `length`. (2 puntos)
4. Declara las estructuras de datos adecuadas para almacenar la información del siguiente supuesto: (1 punto)

Manos de póquer de cinco cartas, en las que cada carta esta caracterizada por un valor numérico (entre 1 y 12) y un palo (rombos, tréboles, corazones y picas).

5. Realiza un programa que lea desde fichero una matriz de valores enteros de tamaño fijo 6x6 y cuente el número de filas que tengan todos los elementos estrictamente positivos. El programa debe de imprimir por pantalla el resultado obtenido. Para ello define los tipos de datos necesarios y modula el programa adecuadamente. (3 puntos)

```
1 2 3 4 5 6
3 4 5 6 5 8
....
```

Ayuda para la realización del examen:

Funciones para el manejo de strings

```
// n° de caracteres de palabra
i = palabra.length();

// inserta palabra en la posición 3 de frase
frase.insert(3, palabra);

// concatena (une) palabra y "hola" y almacena el resultado en
frase
frase = palabra + "hola";

// concatena (añade al final) palabra a frase
frase += palabra;

// borra 7 caracteres de frase desde la posición 3
frase.erase (3,7);

// sustituye (reemplaza) 6 caracteres de frase, empezando en la
posición 1,
// por la cadena palabra
frase.replace (1, 6, palabra);

//busca palabra como una subcadena dentro de frase, devuelve la
posición
//donde la encuentra
i = frase.find(palabra);

//devuelve la subcadena formado por 3 caracteres desde la posición
3 de
//frase
palabra = frase.substr(5,3);

//conversión de string a cadena de caracteres de C
cadena = frase.c_str();
```

1. Función que procesa un fichero:

```
bool ProcesarFichero( string nom)
{
    ifstream f_in;
    ofstream f_out;
    char c;
    bool b_aux;
    int n_lin = 0;
    int n_ca = 0;

    f_in.open (nom.c_str());

    if (!f_in)
        b_aux = false;
    else
    {
        b_aux = true;
        c = f_in.get();
        while ( !f_in.eof() )
        {
            if (c == '\n')
                n_lin ++;
            if ( (( c >= 'a') && ( c<= 'z')) || ( (c >= 'A') && (c <=
'Z') ) )
                n_ca++;
            c = f_in.get();
        }
        f_out.open("salida.txt");
        if ( f_out )
        {
            cout << "Lineas: " << n_lin;
            cout << "\nLetras: " << n_ca;
            b_aux =true;
        }
        else
            b_aux = false;
    }
    return(b_aux);
}
```

2. Convertir formato de fecha

```
string procesarFecha( string str)
{
    string cad_aux,cad;

    cad = str.substr(0,2);
    cad_aux = cad;
    cad = str.substr(3,2);
    cad_aux = cad_aux + " de " + cad;
    cad = str.substr(6,2);
    cad_aux = cad_aux + " de 20" + cad;

    return (cad_aux);
}
```

3. Tamaño de una cadena de forma recursiva

```
int longitud ( string cad)
{
    int i_aux ;

    if (cad == "")
        i_aux = 0;

    else
        i_aux = 1 + longitud(cad.erase(0,1));

    return (i_aux);
}
```

4. Estructura de datos para una mano de 5 cartas

```
const int NUM_CARTAS = 5;

enum lospalos{treboles=1,corazones,rombos,picas};

enum valores{uno=1,dos,tres,cuatro,cinco,seis,siete,ocho,nueva,
diez,once,doce};

struct carta
{
    lospalos palo;
    valores valor_num;
};

typedef carta mano[NUM_CARTAS];
```

5. Programa con matrices.

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

const int MAX =6;
typedef int matriz[MAX][MAX]

void LeerMatriz(matriz m)
{
    ifstream f_in;
    int i,j;
    f_in.open("matriz.txt");
    for (i = 0; i < MAX; i++)
        for (j = 0; j < MAX; j++)
            f_in >> m[i][j];
}

void EscribirMatriz(matriz m)
{
    int i,j;
    for (i = 0; i < MAX; i++)
        for (j = 0; j < MAX; j++)
            cout << m[i][j];
}
```

```
int Positivos( matriz m)
{
    int i,j;
    int n_pos =0;

    for ( i = 0; i< MAX ; i++)
    {
        posi= true;
        for (j = 0; j < MAX ; j++)
            if (m[i][j] <= 0)
                posi = false;
            if (!posi)
                n_pos++;
    }
    return( n_pos);
}

int main()
{
    matriz mat;
    int filas;

    LeerMatriz(mat);
    EscribirMatriz(mat);
    filas = Positivos(mat);

    cout << "El numero de filas positivas es: " << filas << endl;

    system("PAUSE");

    return 0;
}
```