

**FUNDAMENTOS DE PROGRAMACIÓN. 1º ING. INFORMÁTICA (Plan 2000)**  
**2ª Convocatoria. 21 de junio de 2001**

**NORMAS DE EXAMEN:**

(1) La calificación la asignatura depende exclusivamente de este examen; (2) Todas las preguntas tienen la misma valoración; (3) Todas las preguntas se deben realizar usando C++

**I.- Comprensión de código:**

a) Detectad errores de compilación en los siguientes segmentos de código (suponiendo que lo que no aparece aquí está bien). Indicad el número de línea y el tipo de error, sabiendo que puede haber 0 o más errores en cada apartado.

```
(a.1) /*1*/ int sumar (int x, int y)
      /*2*/ {
      /*3*/     res=x+y ;
      /*4*/     return res;
      /*5*/ }
```

```
(a.2) /*1*/ void escribir (string nom)
      /*2*/ {
      /*3*/     ifstream f;
      /*4*/     int x;
      /*5*/
      /*6*/     f.open(nom);
      /*7*/     while (f>>x) cout << x ;
      /*8*/     f.close();
      /*9*/ }
```

```
(a.3) /*1*/ for (double d=0.75; d<5.0; d = d +0.25)
      /*2*/     cout << "d= " << d << endl;
```

```
(a.4) /*1*/ struct complejo
      /*2*/ {
      /*3*/     int real ;
      /*4*/     int imag ;
      /*5*/ }
      /*6*/
      /*7*/ complejo n1, n2 ;
      /*8*/ cout << "Introducir datos n1 y n2:";
      /*9*/ cin >> n1 >> n2;
```

b) Indicad qué devuelve la siguiente función en relación a su entrada:

```
int misterio (int dato)
{
    int res;
    int c, r;
    res = 0;
    do
    {
        c = dato / 10;
        r = dato % 10;
        res = res * 10 + r;
        dato = c;
    }
}
```

```

while ( dato != 0 );
return res;
}

```

**2.-** El desarrollo en serie de Taylor de la función  $e^x$  es:

$$e^x = 1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$$

Realizar una función que evalúe  $e^x$  hasta el término de aproximación  $n$ .

**Prototipo:** double exponente (int x, int n);

**3.-** (a) Declarad y explicad las **estructuras de datos** adecuadas para almacenar la información del siguiente supuesto:

*Una compañía dedicada al transporte marítimo, desea controlar la carga de contenedores en los 20 barcos de su flota. Cada barco posee una bodega con capacidad para 16 contenedores (organizados en 4 filas x 4 columnas).*

*Se desea controlar el nivel de carga de cada barco, almacenando su **nombre** y el **estado de cada posición** de su bodega (“ocupado” o “libre”).*

(b) Escribir una **función** *Control\_Carga(...)* que almacene la información de un fichero en la estructura declarada en el apartado a) y que simultáneamente, calcule y muestre por pantalla el número de contenedores actualmente cargados en la bodega de cada barco y el porcentaje actual de carga de la flota (nº contenedores cargados / nº total contenedores). Ejemplo de archivo: contenido de un archivo “Flota.txt”

```

Ciudad de Mallorca I
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
Cabo de Hornos
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

```

Cada par de líneas corresponde a la información de un barco, en la primera aparece su nombre y en la segunda, la secuencia de contenedores cargados (1) o no (0) en la bodega. Esta secuencia se ajusta al siguiente orden:

Fila1-Col1 Fila1-Col2 Fila1-Col3 Fila1-Col4 Fila2-Col1 Fila2-Col2 ...

**4.-** Escribir una función que dada una cadena de caracteres permita sustituir en una cadena de caracteres todas las ocurrencias de una subcadena por otra. Ejemplo:

- Cadena: *Este problema es mas fácil que el problema anterior.*
- Vieja Subcadena: *problema*
- Nueva Subcadena: *programa*
- Resultado: *Este programa es más fácil que el programa anterior.*

**Prototipo:** void sustituir (string& cad, string vja, string nva);

**5.-** (a) Escribid una función que pida como entrada por teclado parejas de números enteros superiores al valor 1000 y los escriba en un archivo llamado “datos.txt” (no incluir saltos de línea para separar las parejas de valores). (b) Escribid una función que lea el fichero “datos.txt”, pareja por pareja, calcule su suma y lo escriba en un fichero que se llame “suma.txt”. (c) Escribid una función que lea de “datos.txt” las parejas de números y de “suma.txt” el resultado de la suma y los escriba en otro fichero llamado “todo.txt” con el siguiente formato:

NUMERO	NUMERO	RESULTADO
1234	1564	2798
1050	10120	11170

**LABORATORIO DE INTRODUCCIÓN A LA PROGRAMACIÓN (Plan 1993)**  
**2ª Convocatoria. 21 de junio de 2001**

**NORMAS DE EXAMEN:**

(1) La calificación la asignatura depende exclusivamente de este examen; (2) Todas las preguntas tienen la misma valoración; (3) Las preguntas se deben realizar usando *Pascal*

**I.- Comprensión de código:**

a) Detectad errores de compilación en los siguientes segmentos de código (suponiendo que lo que no aparece aquí está bien). Indicad el número de línea y el tipo de error, sabiendo que puede haber 0 o más errores en cada apartado.

(a.1) /\*1\*/ function sumar (x, y : integer) : integer;  
/\*2\*/ begin  
/\*3\*/     res := x+y ;  
/\*4\*/     sumar := res;  
/\*5\*/ end;

(a.2) /\*1\*/ procedure escribir (nom: string[40]);  
/\*2\*/ var f:text;  
/\*3\*/     x: integer;  
/\*4\*/ begin  
/\*5\*/     assign(f,nom);  
/\*6\*/     while (not eof(f))  
/\*7\*/     begin  
/\*8\*/         read(f,x);  
/\*9\*/         write(x);  
/\*10\*/     end;  
/\*11\*/     close(f);  
/\*12\*/ end;

(a.3) /\*1\*/ var d: real;  
/\*2\*/ for d:=0.75 to 5.0 do writeln('d= ',d);

(a.4) /\*1\*/ type complejo = record  
/\*2\*/     preal: integer ;  
/\*3\*/     imag: integer ;  
/\*4\*/     end;  
/\*5\*/  
/\*6\*/ var n1, n2: complejo;  
/\*7\*/ write('Introducir datos n1 y n2:');  
/\*8\*/ readln(n1,n2);

b) Indicad qué devuelve la siguiente función en relación a su entrada:

```
function misterio (dato: integer): integer;  
var res, r, c : integer;  
begin  
  res := 0;  
  repeat  
    c := dato div 10;  
    r := dato mod 10;  
    res := res * 10 + r;  
    dato := c;  
  until ( dato <> 0 );
```

```
misterio := res;
end;
```

2.- El desarrollo en serie de Taylor de la función  $e^x$  es:

$$e^x = 1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$$

Realizar una función que evalúe  $e^x$  hasta el término de aproximación  $n$ .

**Prototipo:** function exponente (x, n: integer): real;

3.- (a) Declarad y explicad las **estructuras de datos** adecuadas para almacenar la información del siguiente supuesto:

*Una compañía dedicada al transporte marítimo, desea controlar la carga de contenedores en los 20 barcos de su flota. Cada barco posee una bodega con capacidad para 16 contenedores (organizados en 4 filas x 4 columnas).*

*Se desea controlar el nivel de carga de cada barco, almacenando su **nombre** y el **estado de cada posición** de su bodega (“ocupado” o “libre”).*

(b) Escribir una **función** *Control\_Carga(...)* que almacene la información de un fichero en la estructura declarada en el apartado a) y que simultáneamente, calcule y muestre por pantalla el número de contenedores actualmente cargados en la bodega de cada barco y el porcentaje actual de carga de la flota (nº contenedores cargados / nº total contenedores). Ejemplo de archivo: contenido de un archivo “Flota.txt”

```
Ciudad de Mallorca I
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
Cabo de Hornos
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

Cada par de líneas corresponde a la información de un barco, en la primera aparece su nombre y en la segunda, la secuencia de contenedores cargados (1) o no (0) en la bodega. Esta secuencia se ajusta al siguiente orden:

Fila1-Col1 Fila1-Col2 Fila1-Col3 Fila1-Col4 Fila2-Col1 Fila2-Col2 ...

4.- Escribir una función que dada una cadena de caracteres permita sustituir en una cadena de caracteres todas las ocurrencias de una subcadena por otra. Ejemplo:

- Cadena: *Este problema es mas fácil que el problema anterior.*
- Vieja Subcadena: *problema*
- Nueva Subcadena: *programa*
- Resultado: *Este programa es más fácil que el programa anterior.*

**Prototipo:** procedure sustituir (var cad:string[80]; vja, nva:string[20]);

5.- (a) Escribid una función que pida como entrada por teclado parejas de números enteros superiores al valor 1000 y los escriba en un archivo llamado “datos.txt” (no incluir saltos de línea para separar las parejas de valores). (b) Escribid una función que lea el fichero “datos.txt”, pareja por pareja, calcule su suma y lo escriba en un fichero que se llame “suma.txt”. (c) Escribid una función que lea de “datos.txt” las parejas de números y de “suma.txt” el resultado de la suma y los escriba en otro fichero llamado “todo.txt” con el siguiente formato:

NUMERO	NUMERO	RESULTADO
1234	1564	2798
1050	10120	11170

# SOLUCIÓN

- 1.-
- (a.1) La variable *res* no está declarada.
  - (a.2) línea 6: No puede usar open con un string, se debe convertir a una cadena de tipo C con `c_str()`
  - (a.3) Código correcto. La declaración de variable dentro del for esta permitida en C++
  - (a.4) línea 5: Falta el ;  
línea 9: deben introducirse los datos de cada complejo campo a campo

b) La función devuelve un entero que es el valor resultante de invertir el orden de las cifras de la variable entera *dato* que se pasa como parámetro.

```
2.-
double exponente (int x, int n)
{
    double suma, termino ;
    int i ;

    termino = 1 ;
    suma = termino;
    for (i=1 ; i<=n ; i++)
    {
        termino = termino * x / i;
        suma += termino;
    }
    return suma;
}
```

```
3.-
// 3.a. Estructuras de datos
const int NBARCOS = 20;
const int MAX = 4;

typedef int TipoBodega[MAX][MAX];

struct Barco
{
    string Nombre;
    TipoBodega Bodega;
};

typedef Barco TipoFlota[NBARCOS];
```

```
//3.b. Funcion Control_Carga()
void Control_Carga (TipoFlota Flota)
{
    ifstream fich;
    int i, j, nbarco = 0;
    int cont, totalcont = 0;

    fich.open("flota.txt");

    if (!fich)
        cout << "Fichero no encontrado" << endl;
```

```

else
{
    while (getline (fich, Flota[nbarco].Nombre))
    {
        cont = 0;
        cout << Flota[nbarco].Nombre << endl;

        for (i=0; i<MAX; i++)
            for (j=0; j<MAX; j++)
            {
                fich >> Flota[nbarco].Bodega[i][j];
                if (Flota[nbarco].Bodega[i][j] == 1)
                    cont++;
            }
        fich.ignore();
        nbarco++;
        totalcont += cont;
        cout << "N° de contenedores: " << cont << endl;
    }
    cout << "Ocupacion: " << (totalcont*100)/(NBARCOS*MAX*MAX) << " %";
}
fich.close();
}

```

4.-

```

void sustituir (string& cad, string vja, string nva)
{
    unsigned int posicion;
    if ( vja != nva ) //si son iguales no se sustituye nada
    {
        posicion = cad.find (vja);
        while ( (posicion>=0) && (posicion<cad.length()) )
        {
            cad.replace (posicion, vja.length(), nva);
            posicion = cad.find (vja);
        }
    }
}

```

5.-

```

#include <fstream.h>
#include <math.h>
#include <stdlib.h>
#include <string>

//apartado a) genera un fichero con los datos leidos
void leer()
{
    int numa, numb;
    ofstream entrada;
    entrada.open("datos.txt");
    cout << "Introduzca la pareja";
    cin >> numa >> numb;
    while (numa != 0 )
    {
        if (numa < 1000 || numb < 1000)

```

```

        cout << "error pareja de entrada";
    else
        entrada << numa << " " << numb<< " ";
    cout << "introduzca la pareja" << endl;
    cin >> numa >> numb;
}
entrada.close();

return;
}

// apartado b) genera un fichero suma
void sumar()
{
    ifstream entrada;
    ofstream salida;
    int numa,numb;
    entrada.open("datos.txt");
    salida.open("suma.txt");
    while ( entrada >> numa)
    {
        entrada >> numb;
        salida << numa + numb<< " ";
    }
    entrada.close();
    salida.close();
    return;
}

// apartado c) genera un fichero con el listado especificado
void listar()
{
    ifstream entradaa,entradab;
    ofstream salida;
    int numa,numb,suma;
    entradaa.open("datos.txt");
    entradab.open("suma.txt");
    salida.open("todo.txt");
    salida << "numero"<<'\t'<<"numero"<<'\t'<<"resultado"<<'\n';
    while (entradaa >> numa)
    {
        entradaa >> numb;
        entradab >> suma;
        salida << numa << '\t' << numb << '\t' << suma << '\n';
    }
    entradaa.close();
    entradab.close();
    salida.close();
    return;
}

```