



EXAMEN DE SISTEMAS INFORMÁTICOS INDUSTRIALES (SOLUCIÓN) (TEORÍA)

SEPTIEMBRE 2013

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

- (a) Suponiendo que **a** y **b** son enteros, y que **b=0**, se puede utilizar cualquiera de estas sentencias: **a=b++**; ó **a=++b**; para hacer que el valor de **a** sea 1. **F**
- (b) Sea **c** una variable tipo **char** que contiene un carácter. Para mostrar el código ASCII del carácter almacenado en **c**, se puede utilizar el siguiente código: **printf("%d", c);** **V**
- (c) La sentencia **continue** dentro de un bucle hace que éste finalice. **F**
- (d) Una función en C no puede llamarse a sí misma. **F**
- (e) Al declarar que una función recibe como parámetro una matriz, se puede dejar vacío el tamaño correspondiente a la primera dimensión de la matriz. Ejemplo: void LeerMatriz(float m[][3]). **V**
- (f) En C no es posible declarar un puntero doble (por ejemplo: **int **p;**), pero sí en C++. **F**
- (g) En C++ una clase derivada no puede heredar de más de una clase base. **F**
- (h) Para que un cliente se comunique con un servidor mediante sockets, el cliente no necesita conocer el puerto del servidor. **F**

(2 puntos)

2. Se desea crear una clase en C++ denominada **CNumeros** que contendrá como variables miembro 2 vectores de números enteros, denominados **menor** y **mayor**, y 2 enteros, denominados **num** y **tam**. Cuando se desee crear un objeto de esta clase, se deberá especificar los valores de **num** y **tam**, de forma que se crearán los vectores de la siguiente forma:

- **tam** indicará el tamaño que tiene que tener cada vector (los vectores deben crearse de forma dinámica).
- La primera posición del vector **menor** tomará el valor de **num** y el resto de posiciones se rellenará con valores consecutivos del valor de **num**. Por ejemplo, si **num=50** y **tam=5**, el vector **menor** tomará el siguiente valor: **50 51 52 53 54**
- La última posición del vector **mayor** tomará el valor de **num** y el resto de posiciones se rellenará con valores consecutivos, empezando desde la primera posición del vector, de forma que el último valor del vector corresponda con **num**. Por ejemplo, si **num=50** y **tam=5**, el vector **mayor** tomará el siguiente valor: **46 47 48 49 50**

Se pide:

- (a) Definir la clase. (0.5 puntos)
- (b) Implementar el constructor que recibe como parámetros el valor de **num** y el valor de **tam**. En caso de que no se especifique ningún parámetro en la llamada al constructor, **num** tomará el valor de 50 y **tam** tomará el valor de 5. (1 punto)
- (c) Implementar el destructor. (0.5 puntos)

Solución:

```
class CNumeros{
private:
    int num;
    int tam;
```



```
        int *menor;  
        int *mayor;  
public:  
    CNumeros(int n=50, int t=5);  
    ~CNumeros(void);  
};
```

```
CNumeros::CNumeros(int n, int t)  
{  
    num=n;  
    tam=t;  
    menor=new int[tam];  
    mayor=new int [tam];  
    for(int i=0;i<tam;i++){  
        menor[i]=num+i;  
        mayor[tam-i-1]=num-i;  
    }  
}
```

```
CNumeros::~~CNumeros(void)  
{  
    delete []menor;  
    delete []mayor;  
}
```

3. Escribir una función denominada **int CiudadMayor (char *nombreFich)** con las siguientes características:

- Recibe como parámetro una cadena de caracteres que contiene el nombre de un fichero. Se considera que dicho fichero ya existe y que en cada línea del fichero hay el nombre de una ciudad (se considera que el nombre de una ciudad sólo consta de una palabra de 20 caracteres como máximo) y el número de habitantes de la ciudad. Ejemplo de fichero:

```
Alicante 334329  
Elche 230354  
Madrid 3293601  
Valencia 798033  
...
```

- La función debe mostrar en pantalla el nombre de la ciudad con mayor número de habitantes.
- Si se produce un error al abrir el fichero, la función devolverá -1. En otro caso devolverá 0.
(2 puntos)

Solución:

```
int CiudadMayor (char *nombreFich)  
{  
    FILE *f;  
    char ciudad[21], ciudadMayor[21];  
    long habitantes,masHabitantes=0;  
  
    f=fopen(nombreFich,"r");
```



```
    if(f==NULL){
        printf("Error: No existe el fichero\n");
        return -1;
    }

    while(!feof(f)){
        fscanf(f,"%s %ld", ciudad, &habitantes);
        if(habitantes>masHabitantes){
            masHabitantes=habitantes;
            strcpy(ciudadMayor,ciudad);
        }
    }

    printf("La ciudad con mas habitantes es: %s\n",ciudadMayor);

    fclose(f);
    return 0;
}
```

4. Escribir la salida del siguiente programa en C++. Si en algún caso no puede determinarse el valor de la variable a imprimir, escribir el símbolo ?.

(2 puntos)



```
#include <iostream.h>

class base
{
protected:
    int i;
    float f;
public:
    base(void);
    virtual ~base(void);
    int funcion1(float y);
    virtual void funcion2(int &j);
};

class derivada : public base
{
private:
    char cadena[30];
public:
    derivada(void);
    ~derivada(void);
    int funcion1(float f);
};

base::base(void)
{
    i = 0;
    cout << "base::base" << endl;
    cout << "Valor de i=" << i << endl;
    cout << "Valor de f: " << f << endl;
}

base::~base(void)
{
    cout << "base::~base " << endl;
}
```

```
int base::funcion1(float y)
{
    cout << "base::funcion1 " << y << endl;
    f = y;
    return(1);
}

void base::funcion2(int &j)
{
    cout << "base::funcion2 " << f << endl;
    j = i;
    i++;
}

derivada::derivada(void) : base()
{
    cout << "derivada::derivada " << endl;
}

derivada::~derivada(void)
{
    cout << "derivada::~derivada" << endl;
}

int derivada::funcion1(float y)
{
    cout << "derivada::funcion1" << endl;
    return base::funcion1(y);
}

void main(void)
{
    int n = 5;
    float r = 8.5;
    base b;
    derivada d;

    b.funcion1(r);
    b.funcion2(n);
    cout << "Valor de n: " << n << endl;
    cout << "Valor de r: " << r << endl;
    d.funcion1(r);
    d.funcion2(n);
    cout << "Valor de n: " << n << endl;
    cout << "Valor de r: " << r << endl;
}
```

Solución:

La salida del programa es:

```
base::base
Valor de i=0
Valor de f: ?
base::base
```



Valor de i=0
Valor de f:?
derivada::derivada
base::funcion1 8.5
base::funcion2 8.5
Valor de n: 0
Valor de r: 8.5
derivada::funcion1
base::funcion1 8.5
base::funcion2 8.5
Valor de n: 0
Valor de r: 8.5
base::~~base
derivada::~~derivada
base::~~base

5. Escribir un programa que pida al usuario que introduzca dos cadenas de caracteres por teclado, almacene las cadenas de caracteres en dos arrays y muestre por pantalla:

- El número de caracteres que coinciden en ambas cadenas empezando desde el primer carácter.
- La subcadena que es igual en las dos cadenas empezando desde el primer carácter.

Para almacenar las cadenas de caracteres utilizar un array de caracteres de tamaño 100. Ejemplo de ejecución (en negrita aparecen las cadenas introducidas por el usuario):

(2 puntos)

```
Introduce cadena 1: Esto es un ejemplo de cadena
Introduce cadena 2: Esto es una frase de ejemplo
Numero de caracteres que coinciden: 10
Subcadena coincidente: Esto es un
```

Solución:

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

void main (void)
{
    char cad1[100],cad2[100],coincidente[100];
    int i;
    printf("Introduce cadena 1:");
    gets(cad1);
    printf("Introduce cadena 2:");
    gets(cad2);

    for(i=0;cad1[i]!='\0' && cad2[i]!='\0';i++){
        if(cad1[i]==cad2[i]) coincidente[i]=cad1[i];
        else break;
    }
    coincidente[i]='\0';
    printf("Numero de caracteres que coinciden: %d\n", i);
    printf("Subcadena coincidente: %s\n", coincidente);
}
```