

PROGRAMACIÓN 1

Ingeniería Informática

27 Junio de 2014

PREGUNTA 1 (1.5 puntos). Calcula el coste temporal del siguiente algoritmo. Exprésalo en forma de tamaño y coste

```
#include <iostream>
using namespace std;

int calcula(int n);

main() {
    int n, cont;

    cout << "\nIntroduce un número positivo:";
    cin >> n;
    cont=n;
    while (cont>=1){
        cout << calcula(n);
        cont=cont-1;
    }
}

int calcula(int n){
    int i, acu;

    acu=0;
    i=1;
    do{
        if (n % i == 0)
            acu=acu+1;
        i=i+1;
    }while (i<n);

    return acu;
}
```

PREGUNTA 2. (1.5 puntos) Realiza un módulo que solicite una serie de números naturales (entre 1 y 9 inclusive) y devuelva el número formado por todos los números introducidos por el usuario. La entrada de datos finaliza cuando el usuario introduce -1 o cuando se han leído 6 números. Ejemplo: para los números 4 8 9 1 6 -1. El módulo devuelve 48916

Importante: En este ejercicio no se podrán utilizar arrays ni registros para almacenar los datos .

PREGUNTA 3. (2 puntos) Diseña un módulo que reciba dos vectores ordenados de manera ascendente de 10 números y devuelva un vector de 20 números formado por los números de los vectores de entrada. Este vector resultante también debe estar ordenado.

PREGUNTA 4. (2 puntos) Escribir una función recursiva que devuelva la suma de los dígitos impares de un número natural. Ejemplo: para el número 321, la función debe devolver 4 y para el número 28, la función debe devolver 0.

Importante: este ejercicio sólo puntuará si se realiza de forma recursiva.

PREGUNTA 5. (1+0.75+1.25 puntos)

Una tienda de videojuegos quiere almacenar información acerca de los productos que vende a sus clientes para poder efectuar campañas de marketing según el perfil del cliente. Para cada videojuego tenemos la siguiente información: código (número entero), nombre, tipo (aventuras, deporte, infantil, educativo), precio y año de lanzamiento. De cada cliente interesa almacenar la siguiente información: código, nombre, dirección y últimos 10 videojuegos que ha comprado. Fijaremos un máximo de 500 videojuegos y 100 clientes.

- a) Diseña las estructuras de datos necesarias para gestionar toda la información.
- b) Diseña el módulo que permite introducir una compra para un determinado cliente (se supone que no es la primera compra del cliente). Ten en cuenta que si ese cliente ya había comprado 10 videojuegos, hay que eliminar de ese cliente el videojuego más antiguo que adquirió para poder introducir el nuevo. Cuando se incorpora una nueva compra, debe hacerse de tal manera que siempre se tenga la información de las 10 últimas compras del cliente.
- c) Diseña un módulo que muestre un listado de todos los videojuegos, indicando nombre, tipo y precio. Este módulo debe devolver al main() el número de videojuegos que hay de cada tipo así como el nombre del videojuego más antiguo.