



Apellido y Nombre

19 / 12 / 2016

DNI: _

Calificación :

1-a.- Escriba una función que busque en una cadena y devuelva en qué dirección de memoria encuentra la primer ocurrencia de uno de los caracteres de otra cadena, de no encontrar devuelve NULL.

1-b.- Escriba una función que determine y devuelva la cantidad de veces que aparece en una cadena uno de los caracteres de otra cadena.

Para ambos ejercicios:

- no debe utilizar subíndices ni puntero más desplazamiento.
- no debe hacer diferencia entre mayúsculas y minúsculas.
- no debe invocar otras funciones.

Se ha cargado en una lista doblemente enlazada los pedidos que han hecho los clientes a lo largo del día. Como el mismo cliente puede haber llamado más de una vez pidiendo uno o varios productos se requiere:

2-a.- una función en que se acumule en un solo nodo los pedidos de cada cliente del mismo producto (la lista no está ordenada).

2-b.- una vez resuelto lo anterior, resolver otra función que ordene la lista por código de cliente/código de producto.

La variable lista estaba con la dirección de algún nodo (con un código de cliente/código de producto en particular). Al final del proceso de acumulación y de ordenamiento, la lista deberá quedar con la dirección del nodo que tenga el mismo código de cliente/código de producto.

3-.- Escriba una `class Persona` con información privada para sus atributos de información: apellido(s) hasta 20 caracteres, nombre(s) hasta 25 caracteres, sexo, altura y un importe de doble precisión.

Resuelva las sobrecargas de las 'funciones miembro' mínimas necesarias e imprescindibles, desarrollándolas fuera de la clase, para el correcto funcionamiento del siguiente trozo de código:

```
void probarPersonas(const char *apel, const char *nom1, char sex1,
                    int edad1, double imp1)
{
    Persona p1(apel, nom1, sex1, edad1, imp1),
            p2(p1),
            p3;
    cout << "Valores iniciales:" << endl
         << "p1" << endl << p1 << endl
         << "p2" << endl << p2 << endl
         << "p3" << endl << p3 << endl << endl;

    p3 = p1++; // modifica la edad
    double imp2 = 90.75;
    Persona p4 = p1 = imp2;

    cout << "Valores finales:" << endl
         << "p1" << endl << p1 << endl
         << "p2" << endl << p2 << endl
         << "p3" << endl << p3 << endl
         << "p4" << endl << p4 << endl << endl;
    if(p1 == p4)
        cout << "Probando la comparacion" << endl
             << p1 << (p1 == p4 ? "Es Identico a" : "Difiere de")
             << endl << p4 << endl;
}
```

EVALUACIÓN EN LABORATORIO

NOTA Para la aprobación se piden correctamente resueltos el punto (a) o (b) de la primera parte y la segunda parte y la correcta declaración de la clase y resueltos los constructores y sobrecarga del "cout" además de otros dos métodos.