



6.- Supongamos que deseamos guardar la información de un mapa estelar, con un total de 1000 estrellas, en forma de grafo, en el que vamos a guardar el nombre de cada estrella y la distancia de cada estrella a las restantes del mapa estelar. En estas condiciones ¿Cuál debería ser la implementación interna del grafo?

- a.- Matrices de adyacencia.
- b.- Listas de adyacencia.
- c.- Matrices dispersas.
- d.- vector circular.

7.- ¿Qué ventaja aporta en la representación estática de las colas, el hecho de hacerla circular?

- a.- Ninguna. De hecho las operaciones se complican.
- b.- El hecho de reaprovechar los huecos que se generan al inicio del vector, si hemos utilizado un 'ini' y un 'fin' en la representación y el hecho de no tener que mover grandes cantidades de información si hemos compactado la información al inicio del vector.
- c.- Ninguna, aunque las operaciones se simplifican considerablemente.
- d.- El hecho de reaprovechar los huecos que se generan al inicio del vector, si hemos utilizado un 'ini' y un 'fin' en la representación y el hecho de no tener que mover grandes cantidades de información, si hemos compactado la información al inicio del vector, aunque la eficiencia global de la estructura disminuye.

8.- Sea la siguiente declaración de tipos

```
typedef string Vec[10];
struct Reg {
    int num;
    Vec dato;
};
typedef Reg Vreg[10];
```

Si *x* es una variable de tipo *vreg*, indicar el tipo resultante de las siguientes expresiones o si son incorrectas y por qué:

* *x*[2].dato _____
x[8] _____
x->dato _____
x[3].dato[5][2] _____
x[0].num _____
x.dato[5] _____

9.- Dado el siguiente programa:

```
void f(int x, int & y);
int main(void)
{
    int x = 1; int y = 2;

    f(y, x);
    y = y + x;
    cout << "x:" << x << "y:" << y;
    return 0;
}
int f(int x, int & y)
{
    x = x + 4;
    y = y + 2;
}
```

¿Qué valores visualizará para *x* e *y*?

X:

Y:

10.- Dado el siguiente vector:

```
typedef char Mchar[7][15][10];
Mchar dato;
```

Y sabiendo que el tamaño de un carácter es 1 byte y que la variable *dato* comienza en la posición de memoria 1000, calcular en qué posición de memoria está *dato*[3][2][5].

**P1.- (3 ptos)**

Sea el siguiente programa en C++:

```
#include<iostream.h>

int f(int n)
{
    int a;
    int res;

    if(n <= 1)
        res = n;
    else
    {
        a = f(n / 10) * 2;
        res = a + n % 10;
    }
    return res;
}

int main(void)
{
    cout << f(1101);
    return 0;
}
```

Realiza una traza del programa y di que mostrará por pantalla.

P2.- (3,5 ptos)

Realizar una función en C++ que devuelva si un árbol binario pasado como parámetro, es o no binario de búsqueda (devolverá `true` si el árbol pasado como parámetro es de búsqueda, `false` si no lo es.)

El prototipo de la función será:

```
bool EsBinarioDeBusqueda (Arbol arb);
```

Escribe el prototipo de los métodos de árboles que vayas a utilizar, así como el código de aquellas funciones auxiliares que creas conveniente.

P3.- (3,5 ptos)

Realizar un nuevo método en C++, para la clase Cola que invierta el contenido de la cola.

```
void Cola::InvertirCola (void)
```

Realizar dos versiones: Una para la representación estática de la cola y otra para la versión dinámica. En ningún caso se podrán utilizar estructuras auxiliares externas y en ambos casos hay que intentar realizar la versión más eficiente posible (es decir, hay que intentar evitar la utilización de los métodos de la clase.)

Escribe en ambos casos la parte privada y las definiciones de tipo necesarias y el código de los métodos auxiliares que creas conveniente.

La representación estática de las colas deberá ser circular.