



Apellidos:

Nombre:

1.- Dada la siguiente declaración de tipos y variables:

```
struct Registro
{
    int x, y;
    float info;
};

typedef Registro * Ptr;

typedef Ptr Matriz [2][3];

Matriz x;
```

Di si son correctas y que se obtiene en las siguientes expresiones. O si son incorrectas y por qué.

<code>x[1][2]</code>	<code>_Correcto_Ptr</code>	_____
<code>*x[1]</code>	<code>_Correcto_Ptr</code>	_____
<code>*x[1][2]</code>	<code>_Correcto_Registro</code>	_____
<code>*x[1][2].info</code>	<code>_Incorrecto_x[1][2] es un puntero y no se puede acceder con el punto a los campos de Registro</code>	_____
<code>**x.info</code>	<code>_Incorrecto_ (**x) es un puntero y no se puede acceder con el punto a los campos de Registro</code>	_____
<code>x[1][2]->x</code>	<code>_Correcto_int</code>	_____

2.- Sea la siguiente declaración de tipos y de variables:

```
typedef double Vect[5];

struct reg
{
    Vect dat;
    float b;
    int a;
};

typedef reg MT[4][10];

MT x;
```

Sabiendo que la variable 'x', comienza en la posición de memoria 1000, que el tamaño de un int es de 2 bytes, el de un float es de 4 bytes, el de un puntero es de 6 bytes y el del double 7, determina la posición de memoria en la que se encontrará la siguiente información:

`x[2][4].dat[3]`

$$\text{sizeof (reg)} = 5 * \text{sizeof (double)} + \text{sizeof (float)} + \text{sizeof (int)} = 5 * 7 + 4 + 2 = 41$$

$$\text{sizeof (reg[10])} = 10 * \text{sizeof (reg)} = 10 * 41 = 410$$

$$m = m_0 + 2 * \text{sizeof (reg[10])} + 4 * \text{sizeof (reg)} + 3 * \text{sizeof (double)} =$$

$$= 1000 + 2 * 410 + 4 * 41 + 3 * 7 = 1000 + 820 + 164 + 21 = 2005$$

3.- Dada la siguiente gramática:

$\langle E \rangle := \langle S \rangle = \langle N \rangle$
 $\langle S \rangle := \langle N \rangle + \langle N \rangle \mid \langle N \rangle - \langle N \rangle$
 $\langle N \rangle := + \langle D \rangle * \langle N \rangle \mid - \langle D \rangle * \langle N \rangle \mid \langle D \rangle * \langle D \rangle$
 $\langle D \rangle := 0 \mid 6$

Di qué frases son correctas y cuales son incorrectas según esta gramática:

	V / F		V / F	
a) $0 * 0 = + 0 * + 0 * 0$	<u> F </u>		d) $0 * 6 + 0 * 0 * 6 = 6 * 6$	<u> V </u>
b) $0 + 0 = + 0 * + 0 * 0$	<u> F </u>		e) $- 6 * 0 + 0 * 0 = + 6 * 6 * 6$	<u> V </u>
c) $0 * 0 + 6 * 6 = 0 * 0$	<u> V </u>			

Nota: Cada apartado correcto puntúa 0.2. Cada apartado incorrecto resta -0.1.

4.- Dada la siguiente declaración de variables:

```
float c = 1.0;
unsigned int n = 0;
double b = 4.0;
unsigned short d = 2;
int a[3] = {9, 16, 25}
```

Y sabiendo que el prototipo de la función 'sqrt' es:

```
double sqrt (double);
```

Evalua paso a paso la siguiente expresión:

$b / a[n+2] * a[n+1] + a[n+2] * 2 / a[n+2] < 10 / \text{int}(\text{sqrt}(a[0])) \ \&\& \ a[n+2] \% d * c <= n$
 $4,0 / 25 * 16 + 25 * 2 / 25 < 10 / \text{int}(3,0) \ \&\& \ 25 \% 2 * 1,0 <= 0$
 $0,16 * 16 + 50 / 25 < 10 / 3 \ \&\& \ 1 * 1,0 <= 0$
 $<= 0 \quad 2,56 + 2 < 3 \ \&\& \ 1,0$
 $4,56 < 3 \ \&\& \ \text{false}$
 $\text{false} \ \&\& \ \text{false}$



Apellidos:

Nombre:

5.- Dado un árbol binario de altura 20, indica...

a.- ¿Cuál es el número máximo de nodos del árbol?

$$2^{20} - 1 = 1048575$$

b.- ¿Cuál es el número mínimo de nodos del árbol?

20

c.- ¿Cuál es el número máximo de hojas?

$$2^{20-1} = 524288$$

d.- ¿Cuál es el número mínimo de hojas?

1

Nota: $2^{10} = 1024$

6.- Dado el siguiente vector {7, 5, 11, -6, 7, 0} indica el número de comparaciones de elementos que se realiza para ordenar el vector en cada uno de los siguientes métodos:

Inserción con centinela	(a) 14	(b) 18	(c) 21	(d) 23
Selección	(a) 10	(b) 15	(c) 19	(d) 21
Quick-Sort	(a) 10	(b) 13	(c) 16	(d) 19

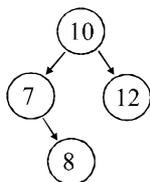
7.- En una lista con punto de interés implementada de forma dinámica (simplemente enlazada) indicar cuál de las siguientes operaciones tiene un coste lineal en el mejor de los casos:

- (a) Añadir un elemento.
- (b) Eliminar un elemento.
- (c) Constructor de la clase.
- (d) Constructor de copia.**

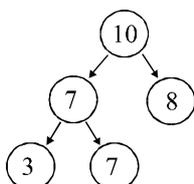
8.- Una lista con punto de interés se puede implementar mediante dos pilas. Como una cola no es más que un tipo especial de lista, podemos plantearnos la implementación de una cola mediante dos pilas. ¿Sería conveniente esta posibilidad atendiendo a su eficiencia?

- (a) Sí, como con listas.
- (b) No, porque no podría implementarse con un coste constante uno de los métodos principales: o bien el método encolar, o bien el método desencolar.**
- (c) No porque ni el método encolar, ni el método desencolar podrían implementarse con un coste constante.
- (d) Sí, incluso mejora la eficiencia de la implementación dinámica.

9.- Dado los siguientes árboles binarios, indica cuál de las afirmaciones es cierta:



- (a) Es un árbol binario de búsqueda.**
- (b) Es un montículo.
- (c) No es un árbol binario de búsqueda porque no es completo.
- (d) No es un árbol binario de búsqueda porque no es lleno.



- (a) Es un árbol binario de búsqueda.
- (b) Es un montículo.**
- (c) No es un árbol binario de búsqueda porque no es completo.
- (d) No es un montículo porque tiene claves repetidas.

10.- Supuesto un problema en el que se necesita utilizar un grafo con 100 nodos en los que guardamos valores enteros y 3500 arcos, di cuál es el coste espacial de la representación mediante...

a.- Matrices de adyacencia.

$$100 * \text{int} + 100 * 100 * \text{bool} = \\ = 100 * 2 + 10000 * 1 =$$

b.- Listas de adyacencia.

$$100 * (\text{int} + \text{ptr}) + 3500 * (\text{int} + \text{ptr}) = \\ = 100 * (2 + 6) + 3500 * (2 + 6) =$$

= 10200 bytes

= 100 * 8 + 3500 * 8 = 28800 bytes

Nota: el tamaño de un bool 1 byte, el de un int es de 2 bytes, el de un float es de 4 bytes, el de un puntero es de 6 bytes y el del double 7



P.1.- Realizar un método que determine si en una pila existen elementos repetidos. Suponer que la pila tiene una implementación dinámica.

Se valorará negativamente la utilización de otros métodos de la clase, así como la utilización de estructuras de datos auxiliares.

```
bool Pila::Repetidos (void)
{
    Puntero p_aux1, p_aux2;
    bool repes = false;

    if ( (cima != NULL) && (cima->sig != NULL) )
    {
        p_aux1 = cima;
        while ( (p_aux1->sig != NULL) && (!repes) )
        {
            p_aux2 = p_aux1->sig;
            while ( (p_aux2 != NULL) && (!repes) )
            {
                if (p_aux1->info == p_aux2->info)
                    repes = true;
                else
                    p_aux2 = p_aux2->sig;
            }
            p_aux1 = p_aux1->sig;
        }
    }
    return repes;
}
```

P.2.- Realizar una función a la que se le pase un árbol binario de búsqueda y nos devuelva el valor contenido en la hoja de menor profundidad.

La interfaz de clase árbol binario de búsqueda es la siguiente:

```
class ABB //Arbol Binario de Busqueda
{
    public:
        ...

    private:
        ...
};

bool HojaMenor (const Arbol & a, int & niv, Valor & x)
{
    int n_izq, n_der;
    Valor x_izq, x_der;
    bool éxito, e_izq, e_der;

    if (a.ABBVacio () )
        éxito = false;
    else
    {
        éxito = true;

        // Si es una hoja
        if (a.HijoIzdo ().ABBVacio () && a.HijoDcho ().ABBVacio () )
        {
            a.Informacion (x);
            niv = 1;
        }
        else
        {
            e_izq = HojaMenor (a.ArbolIzdo (), n_izq, x_izq);
            e_der = HojaMenor (a.ArbolDcho (), n_der, x_der);

            if (!e.der)
            {
                niv = n_izq + 1;
                x = x_izq;
            }
            else
            {
                if (!e.izq)
                {
                    niv = n_der + 1;
                    x = x_der;
                }
                else
                {
                    if (n_izq < n_der)
                    {
                        niv = n_izq + 1;
                        x = x_izq;
                    }
                    else
                    {
                        niv = n_der + 1;
                        x = x_der;
                    }
                }
            }
        }
    }
    return éxito;
}
```

P.3.- Dado el siguiente programa, realiza la traza:

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

int f (int, int &);

int main (void)
{
    int r;
1   int x = 10;
2   int y = 3;

3   r = f (x, y);

    cout << r << x << y;

    return 0;
}
```

```
int f (int a, int & b)
{
    int r;

    if (a == b)
4      r = 1;
    else
        if (a % b == 0)
5          r = 0;
        else
        {
6          b = b + 1;
7          r = f (a, b);
        }

    return r;
}
```

	x	y	r									
1	10	?	?									
2	10	3	?	a ₁	b ₁ /y	r ₁						
3	10	3	?	10	3	?						
6 ₁	10	4	?	10	4	?	a ₂	b ₂ /y	r ₂			
7 ₁	10	4	?	10	4	?	10	4	?			
6 ₂	10	5	?	10	5	?	10	5	?	a ₃	b ₃ /y	r ₃
7 ₂	10	5	?	10	5	?	10	5	?	10	5	?
5 ₃	10	5	?	10	5	?	10	5	?	10	5	0
7 ₂	10	5	?	10	5	?	10	5	0			
7 ₁	10	5	?	10	5	0						
2	10	5	0									