



Apellidos:

Nombre:

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

```
typedef float * Ptr;
typedef Ptr Vector [3];
struct Registro
{
    Ptr punt;
    Vector info;
};
Registro 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 y el de un puntero es de 8 bytes, determina la posición de memoria en la que se encontrará la siguiente información:

$$\begin{aligned}
 x.info[2] &= m0 + \text{sizeof (Ptr)} + 2 * \text{sizeof (Ptr)} = \\
 &= 1000 + 8 + 2 * 8 = 1024
 \end{aligned}$$

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

```
typedef int Vect1[3];
struct Reg1
{
    Vect1 a1;
    float b1;
};
typedef Reg1 Vect2[10];
struct Reg2
{
    Vect2 a2;
    string b2;
};
Reg2 x;
```

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

x.a2[10]	Cor /Inc	Se sale de rango
x.a2[8].a1[2]	Cor/ Inc	int
*x.a2[8].a1	Cor/ Inc	int
x.a2[0][1]	Cor /Inc	a2 no es una matriz
x.b2[0]	Cor/ Inc	char
x.a2.a1[2]	Cor /Inc	a2 es un vector
*x.a2.a1[2]	Cor /Inc	a2 es un vector y a1[2] es un int y no un puntero por

| **__lo que no se puede usar *__**



Apellidos:

Nombre:

3.- Dada la siguiente gramática:

```
<E> ::= <Q><O><Q>
<O> ::= + | - | <
<Q> ::= <D>.<D> | <D><D>
<D> ::= 0 | 9
```

Determina si las siguientes frases que siguen esta gramática son correctas o incorrectas:

9.9-0.0	Correcto	Incorrecto
1.3-21	Correcto	Incorrecto
-9+0.9	Correcto	Incorrecto

99<00	Correcto	Incorrecto
0<99	Correcto	Incorrecto

4.- Indica qué se muestra por pantalla cuando se ejecute el siguiente código:

```
int i, j;

j = 1;

for (i = 1; i < 6; i++)
    if (i % 2 == 0)
    {
        j = i;
        while (j < 5)
        {
            cout << j;
            j = j + 1;
        }
    }
    else
        if (i == j)
            cout << -j;
```

Pantalla:

```
-12344-5
```

5.- Dada la siguiente declaración de variables:

```
float a = 4.99;
int b = 5;
double c = 2.5;
unsigned short d = 3;
```

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

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

Evalua paso a paso la siguiente expresión:

```
int (a) / (b / sqrt (4) * int (c) ) > 4 / 5 && 3 + 5 % d - 5 >= 0
4 / (5 / 2.0 * 2 ) > 4 / 5 && 3 + 5 % 3 - 5 >= 0
4 / ( 2.5 * 2 ) > 0 && 3 + 2 - 5 >= 0
4 / 5.0 > 0 && 0 >= 0
0.8 > 0 && true
true && true
true
```

6.- Di qué muestra por pantalla el siguiente programa:

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

int main (void)
{
    ofstream f1;
    ifstream f2;
    int i;
    char c;

    f1.open ("datos.txt");
    for (i = 8; i < 18; i++)
        f1 << i << '-';
    f1.close ();

    f2.open ("datos.txt");
    for (i = 1; i < 10; i++)
    {
        f2 >> c;
        cout << c;
    }
    f2.close ();

    return 0;
}
```

Pantalla:

8-9-10-11

7.- Dada la siguiente declaración de variables:

```
int a = 1;
int *b, *c, *d;
```

Di qué se va mostrando por pantalla tras cada ejecución del código, sabiendo que la dirección de memoria de la variable 'a' es 1050:

```
b = &a;
c = b;
d = new int;
*d = a;
cout << *c << *d << *b << a;
```

Pantalla

1 1 1 1

```
*b = *c + *d;
cout << *c << *d << *b << a;
```

Pantalla

2 1 2 2

```
b = d;
cout << *c << *d << *b << a;
```

Pantalla

2 1 1 2

```
a = a * (*c) + *d;
cout << *c << *d << *b << a;
```

Pantalla

5 1 1 5



Apellidos:

Nombre:

8.- Dado el siguiente programa, realiza la traza:

```
#include <iostream.h>

int F (int);

int main(void)
{
    int a, r;

    a = 1;

    r = F (a);

    return 0;
}
```

```
int F (int a)
{
    int r;

    if (a < 3)
    {
        r = f (a + 1);
        r += f (a + 2);
    }
    else
        r = a;

    return r;
}
```

	<i>a</i>	<i>r</i>							
1	1	?	<i>a</i> ₁	<i>r</i> ₁					
2	1	?	1	?	<i>a</i> ₂	<i>r</i> ₂			
3 ₁	1	?	1	?	2	?	<i>a</i> ₃	<i>r</i> ₃	
3 ₂	1	?	1	?	2	?	3	?	
5 ₃	1	?	1	?	2	?	3	3	
3 ₂	1	?	1	?	2	3	<i>a</i> ₄	<i>r</i> ₄	
4 ₂	1	?	1	?	2	3	4	?	
5 ₄	1	?	1	?	2	3	4	4	
4 ₂	1	?	1	?	2	7			
3 ₁	1	?	1	7	<i>a</i> ₅	<i>r</i> ₅			
4 ₁	1	?	1	7	3	?			
4 ₅	1	?	1	7	3	3			
4 ₁	1	?	1	10					
2	1	10							