



Apellidos:

Nombre:

1.- Dada la declaración de tipos

```
typedef float V1[3];

struct R1
{
    V1 c1;
    double c2;
};

typedef R1 V2[10];
```

```
typedef R1 M1[4][10];

struct R2
{
    V2 c3;
    M1 c4;
};

typedef R2 V3[100];
```

Si 'x' es una variable de tipo 'V3', y sabiendo que la variable 'x' empieza en la posición 1000, determina la dirección de memoria en la que se encuentra la información $x[10].c4[2][4].c2$.

Nota: El tamaño de un char es 1 byte, el tamaño de un int es 2 bytes, el tamaño de un float son 3 bytes, el tamaño de un double es 6 bytes y el tamaño de un puntero es 4 bytes.

$x[10].c4[2][4].c2 =$

```
sizeof(V1) = 3 * sizeof(float) = 9
sizeof(R1) = sizeof(V1) + sizeof(double) = 15
sizeof(V2) = 10 * sizeof(R1) = 150
sizeof(M1) = 4 * 10 * sizeof(R1) = 600
sizeof(R2) = sizeof(V2) + sizeof(M1) = 750
```

```
pos(x) = 1000
pos(x[10]) = 10 * sizeof(R2) + pos(x) = 7500 + 1000 = 8500
pos(x[10].c4) = sizeof(V2) + pos(x[10]) = 150 + 8500 = 8650
pos(x[10].c4[2][4]) = 2 * sizeof(R1[10]) + 4 * sizeof(R1) + pos(x[10].c4)
= 2 * 150 + 4 * 15 + 8650 = 300 + 60 + 8650 = 9010
pos(x[10].c4[2][4].c2) = sizeof(V1) + pos(x[10].c4[2][4]) = 9 + 9010 = 9019
```

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

```
typedef int * PtInt;
typedef float * PtFloat;
typedef float Mat[10];
typedef Mat Vec[5];

struct Str
{
    PtInt v0; // int *v0;
    PtFloat v1; // float *v1;
    Vec v2;
};

typedef Str X[5][10];

X x;
```

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

```
x[0][9][3][4].v2
    Cor/Inc ___ x es un array de 2 Dimen ___

*(x[1][1].v0)
    Cor/Inc ___ int _____

*x[0][9].v2
    Cor/Inc ___ Mat _____

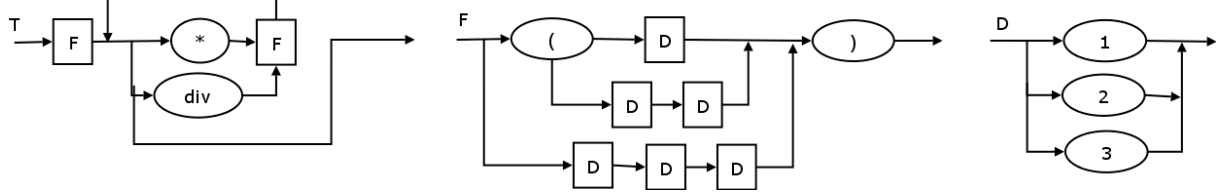
x[0][9].v2[0][1]
    Cor/Inc ___ float _____

x[0][9]->v1
    Cor/Inc ___ x[0][9] no es un puntero ___

x[0][9].v2[3].v1
    Cor/Inc ___ x[0][9].v2[3] es un vector, NO
    un registro. _____

x[0][9].(*v0)
    Cor/Inc ___ Tras el operador "." debe ___
    aparecer simplemente el ___
    nombre del campo _____
```

3.- Dada la siguiente gramática:



Determina si las siguientes frases son correctas o incorrectas según la anterior gramática:

123)	Correcto	Incorrecto
((1)*(2))*(3)	Correcto	Incorrecto
(21)*(21)div321)	Correcto	Incorrecto

(21)div(123)	Correcto	Incorrecto
132div(5)	Correcto	Incorrecto

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

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

void F (int & b, int & a, int c)
{
    b = a;
    a = b + c;
    c = a + b;

    return;
}
```

```
int main (void)
{
    int a = 3, b = 7;
    int c = a - b;

    F (a, b, c);
    cout << a << b << c;

    return 0;
}
```

Pantalla:

```
7 3 -4
```

5.- Evalúa paso a paso la siguiente expresión:

```
!( !false || 12 <= 9/2.0 * 3) && 13 % 10 * int (true) >= int (3 * (3/2) ) || 1 + 1 * 1/2 == 1.5

!( !false || 12 <= 9/2.0 * 3) && 13 % 10 * int (true) >= int (3 * 1 ) || 1 + 1 * 1/2 == 1.5

!( true || 12 <= 9/2.0 * 3) && 13 % 10 * int (true) >= int (3) || 1 + 1 * 1/2 == 1.5

!( true || 12 <= 4.5 * 3) && 13 % 10 * 1 >= 3 || 1 + 1 * 1/2 == 1.5

!( true || 12 <= 13.5) && 13 % 10 * 1 >= 3 || 1 + 1 * 1/2 == 1.5

!( true || true) && 13 % 10 * 1 >= 3 || 1 + 1 * 1/2 == 1.5

false && 13 % 10 * 1 >= 3 || 1 + 1 * 1/2 == 1.5

false && 3 * 1 >= 3 || 1 + 1 * 1/2 == 1.5

false && 3 >= 3 || 1 + 1 * 1/2 == 1.5

false && 3 >= 3 || 1 + 1/2 == 1.5

false && 3 >= 3 || 1 + 0 == 1.5

false && 3 >= 3 || 1 == 1.5

false && true || false

false || false

false
```



Apellidos:

Nombre:

6.- Dado el siguiente código:

```
i = 10;
cin >> n;
j = n;
while (j >= 0 && j <= 10)
{
    if (j % 2 == 1)
    {
        j--;
        i = i + j;
    }
    else
    {
        j--;
        i = j - i;
    }
}
```

Realiza dos versiones, una empleando bucles 'do...while' y otra con bucles 'for', que realicen exactamente la misma tarea para cualquier valor de 'n'.

Versión con bucles 'do...while'

```
i = 10;
cin >> n;
j = n;

if ( (j <= 10) && (j >= 0) )
do
{
    if (j % 2 == 1)
    {
        j--;
        i = i + j;
    }
    else
    {
        j--;
        i = j - i;
    }
}
while (j >= 0 && j <= 10);
```

Versión con bucles 'for'

```
i = 10;
cin >> n;
j = n;

if ( j <= 10 )
for (j = n; j >= 0; j--)
{
    if ( j % 2 == 1 )
        i = i + (j - 1);
    else
        i = (j - 1) - i;
}
```

7.- Dado el siguiente código:

```
1    int *p, *q, *r, *s;
2    int x;

3    x = 7;
4    p = &x;
5    x = 2;
6    q = p;
7    r = new int;
8    s = r;
9    *s = x + *q;
10   r = q;
11   *q = *p + 3;

12   cout << *p << *q << *r << *s << x << endl;
```

Determina si es o no correcto.

Si es correcto, di lo que muestra por pantalla. Si es incorrecto di qué línea o líneas son incorrectas y explica brevemente el error

El código es correcto y lo que se muestra por pantalla es:

5 5 5 4 5



Apellidos:

Nombre:

8.- Realiza la traza del siguiente programa,:

```

#include <iostream>
using namespace std;

int A (int, int);

int main (void)
{
(1)   int a = 4;
(2)   int b = 4;

(3)   b = A (a, b);
      cout << a << b;

      return 0;
}

int A (int b, int a)
{
      int c;

(4)   c = 1;
      if (a > 1)
      {
(5)       c = A (b, a - 1);
          if (a % 2 == 1)
(6)           c = b / c + a;
          else
(7)           c = c + 1;
      }
      return c;
}

```

	a	b												
1	4	?												
2	4	4	b ₁	a ₁	c ₁									
3	4	4	4	4	?									
4 ₁	4	4	4	4	1	b ₂	a ₂	c ₂						
5 ₁	4	4	4	4	1	4	3	?						
4 ₂	4	4	4	4	1	4	3	1	b ₃	a ₃	c ₃			
5 ₂	4	4	4	4	1	4	3	1	4	2	?			
4 ₃	4	4	4	4	1	4	3	1	4	2	1	b ₄	a ₄	c ₄
5 ₃	4	4	4	4	1	4	3	1	4	2	1	4	1	?
4 ₄	4	4	4	4	1	4	3	1	4	2	1	4	1	1
5 ₃	4	4	4	4	1	4	3	1	4	2	1	<-	--	-
7 ₃	4	4	4	4	1	4	3	1	4	2	2			
5 ₂	4	4	4	4	1	4	3	2	<-	--	-			
6 ₂	4	4	4	4	1	4	3	5						
5 ₁	4	4	4	4	5	<-	--	-						
7 ₁	4	4	4	4	6									
3	4	6	<-	--	-									