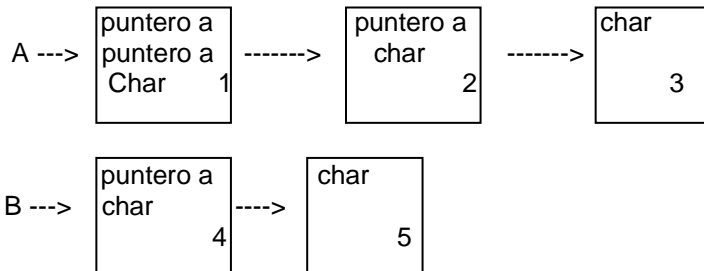


Apuntes

C10-Compatibilidad y Conversiones

1. Se tienen las siguientes variables:



en C

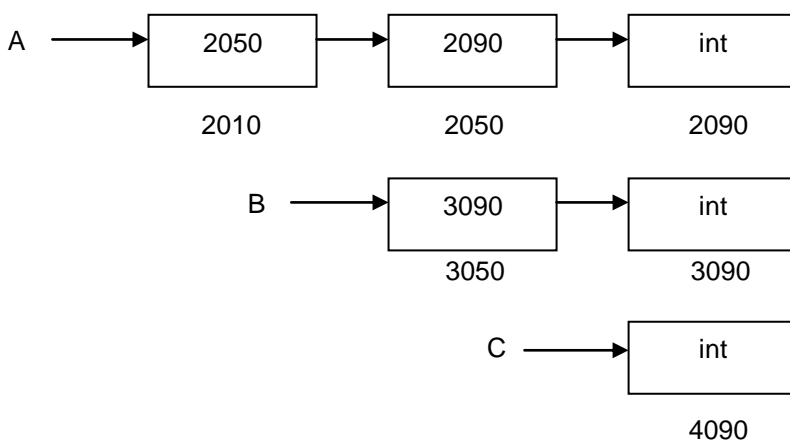
Se pide: (indicando cantidad de “desreferencings” en cada caso)

- Hacer que 1 apunte a 4
- Hacer que 2 se copie en 4
- Hacer que 4 se copie en 2
- Hacer que 2 apunte a 5
- Hacer que 4 apunte a 3
- Copiar 3 en 5
- Copiar 5 en 3
- Copiar 1 en 4

¿Cómo serían las instrucciones si el lenguaje tuviese las conversiones en forma implícita del lado derecho de las sentencias?

2. Dadas las siguientes definiciones en C:

```
int ** A;
int * B;
int C;
```



Indicar para cada conjunto de asignaciones

- si es posible realizar la asignación
- de ser posible, cuántos “desreferencings” se producen en cada una
- cómo quedarían las celdas de memoria luego de ejecutarse las asignaciones.

- a. `A:=3;`
`B:=A;`
- b. `C:=1;`
`B:=&C;`
`A:=&B;`
- c. `** A:=5;`
`B:=*A;`
- d. `C:=3;`
`A:=C;`
- e. `* B:=5;`
`** A:=*B;`
- f. `C:=3`
`* B:=C;`
`A:=&B;`
`C:=**A;`
- g. `*B:=2;`
`* A:=B;`

3. Sean las siguientes declaraciones escritas en C:

```
int a, b;
int * c,d;
int ** e;
```

Indique a que valores referencian las variables después de las asignaciones que siguen. En caso que la asignación sea inválida, indique por qué? En caso que sea válida indique el valor final y la cantidad de referencin en cada caso.

- a) `a := 1;` `c := a;` `b := c;`
- b) `b := 5;` `d := &b;` `e := &d;` `a := **e`
- c) `a := 3;` `*d := a;` `e := a;` `b := e;`
- d) `c := 1;` `e := c;` `a := e;`

4. Dado el siguiente segmento de código Ada indique para cada asignación (señalizada con números):
¿Qué errores se producen en tiempo de compilación y por qué?
Realice las conversiones necesarias para llevar a cabo la asignación.

```
procedure Conversiones is
```

```
  subtype smallFloat is Float range 0.0..10.0;
  type smallSmallFloat is new Float range 0.0..10.0;
```

```
  area, coeficiente, promedio, total: Float;
  suma, cantidad: Integer;
  caracter: character;
  pi: constant:= 3.14159_26536;
```

```

radio: smallFloat;
radio1: smallamallFloat;

begin
  cantidad:= coeficiente +promedio*2.85;      -- 1
  suma:= (cantidad-15.7) + suma;              -- 2
  promedio:= suma/cantidad;                   -- 3
  cantidad:= caracter;                         -- 4
  area:= pi*(radio*radio);                    -- 5
  area:= smallsmallFloat(pi)*(radio1*radio1); -- 6
end;
```

5. Dadas las siguientes definiciones de tipos, subtipos, y declaración de variables, pruebe y analice para cada una de los puntos referenciados

```
with Gnat.IO; use Gnat.IO;
```

```
procedure Conversiones is
```

```
type COLOR is (WHITE, RED, GREEN, BLUE, YELLOW, BROWN, BLACK);
```

```
type RGB is new COLOR range RED..BLUE;
```

```
type SMALLINT is new INTEGER range -1000..1000;
```

```
subtype PAGENUMBER is SMALLINT range 0..100;
```

```
windowColor: COLOR;
```

```
lineColor: RGB;
```

```
page: PAGENUMBER;
```

```
interval: SMALLINT;
```

```
length,marginRight: FLOAT;
```

```

lineColor:= RGB(windowColor);                --1
page:= PAGENUMBER(INTEGER(length)/25);        --2
interval:= SMALLINT(page);                    --3
page:= PAGENUMBER(interval);                  --4
marginRight:= FLOAT(interval*page-PAGENUMBER(length/0.698)); --5
interval:= page;                              --6
interval:= PAGENUMBER(INTEGER(length)/25);    --7
page:= 100;                                   --8
interval:= page + 1;                          --9
interval:= page + 10000;                      --10
```

```
End Conversiones;
```

- Los tipos de conversiones efectuadas (numérica o de tipos derivados).
 - En qué casos las conversiones no producen error en tiempo de compilación, pero sí en tiempo de ejecución
6. UNIONES : Idem anterior para el siguiente fragmento :

```

#include <stdio.h>
#include <string.h>
#include <alloc.h>
#include <process.h>
```

```

int main(void)
{
    struct uno {
        int i;
        char* f;
    } a;

    union dos {
        int i;
        char* f;
    } b,c;

    union tres {
        int i;
        char* f;
    } d;

    a.i = 4000 ;
    a.f = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" ;
    b.i = 5000 ;
    b.f = "bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb" ;
    d = b ;
    return 0;
}
-----1
-----2
-----3
-----4
-----5

```

7. Para el siguiente ejemplo en el lenguaje Pascal, analice las asignaciones en los puntos indicados y concluya respecto de la seguridad de las uniones en este lenguaje.

```

type
    figura =(circulo,triangulo,rectangulo);
    objeto = record
        case forma : figura of
            circulo: (diametro : real);
            triangulo: (ladoizq : integer;
                       ladoder : integer;
                       angulo : real);
            rectangulo: (lado1 : integer;
                       lado2 : integer)
        end;
var
    cuerpo : objeto;
    lado : integer;
begin
    Readln (cuerpo.forma);
    cuerpo.forma := rectangulo;
    cuerpo.lado1:= 1;
    cuerpo.diametro :=2.73;
    lado:=cuerpo.lado1;
-----1
-----2
-----3
-----4
-----5

```

8. Idem anterior para el siguiente ejemplo en el lenguaje ADA

```

With Gnat.IO; use Gnat.IO;
procedure reg_var is
    type FIGURA is (circulo,triangulo,rectangulo);
    type objeto (FORMA: FIGURA:=triangulo) is
        record
            case FORMA is

```

```

        when circulo => diametro : float;
        when triangulo => ladoizq : integer;
            ladoder : integer;
            angulo : float;
        when rectangulo => lado1 : integer;
            lado2 : integer;

        end case;
    end record;

    a: objeto;           -- registro libre
    c: objeto (circulo);  -- registro congelado
    r: objeto (FORMA => rectangulo); -- registro congelado
    r1: objeto (FORMA => rectangulo);
    t: objeto (FORMA => triangulo);
    lado:integer;
begin
--1    a:= (FORMA => triangulo, ladoizq => 10, ladoder => 20, angulo => 45.00);
--2    t:= a;
--3    t:= (FORMA => triangulo, ladoizq => 43, ladoder => 53 );
--4    t:= (FORMA => triangulo, 45,12,0.45);
--5    c:= (FORMA => rectangulo, lado1 => 100, lado2 => 50);
--6    c:= (diametro => 35.4 );
--7    c:= (FORMA => circulo, diametro =>4.9);
--8    r:= (FORMA => rectangulo, lado1 => 100, lado2 => 50);
--9    r1 := r;
--10   lado:=r.lado1;
--11   Put_Line ("lado es: ");Put(lado);

end reg_var;

```

9. Indique los resultados / errores en compilación / ejecución del siguiente código JAVA

```

public class ConversionTipos {

    public static void main(String args[]) {
        byte b;
        int i = 257;
        double d = 323.142;
        double d2;
        float f = 5.65f;
        char c = 'c';
        short s = 1024;

        b = (byte) i;
        System.out.println(b);
        i = (int) d;
        System.out.println(i);
        b = (byte)d;
        System.out.println(b);
        i = (int)f;
        System.out.println(i);
        i = c;
        System.out.println(i);
        f = c;
        d = c;
    }
}

```

```
    i = b * c;  
    System.out.println(i);  
    d2 = (f*b) + (i/c) - (d*s);  
    System.out.println(d2);  
}
```