



1. Dentro de la función sólo vamos a mirar los tres bits de menor peso (0x1, 0x2 y 0x3), por lo que no nos tenemos que preocupar del resto. Obtenemos mediante divisiones sucesivas los tres primeros restos de cada parámetro y lo mostramos en orden invertido:

23	48	421	124
$\begin{array}{r} 23 \quad   \quad 2 \\ 1 \quad 11 \quad   \quad 2 \\ \quad 1 \quad 5 \quad   \quad 2 \\ \quad \quad 1 \quad 2 \end{array}$	$\begin{array}{r} 48 \quad   \quad 2 \\ 0 \quad 24 \quad   \quad 2 \\ \quad 0 \quad 12 \quad   \quad 2 \\ \quad \quad 0 \quad 6 \end{array}$	$\begin{array}{r} 421 \quad   \quad 2 \\ 1 \quad 210 \quad   \quad 2 \\ \quad 0 \quad 105 \quad   \quad 2 \\ \quad \quad 1 \quad 52 \end{array}$	$\begin{array}{r} 124 \quad   \quad 2 \\ 0 \quad 62 \quad   \quad 2 \\ \quad 0 \quad 31 \quad   \quad 2 \\ \quad \quad 1 \quad 15 \end{array}$
111	000	101	100
rwX	---	r-x	r--

Escribirá:

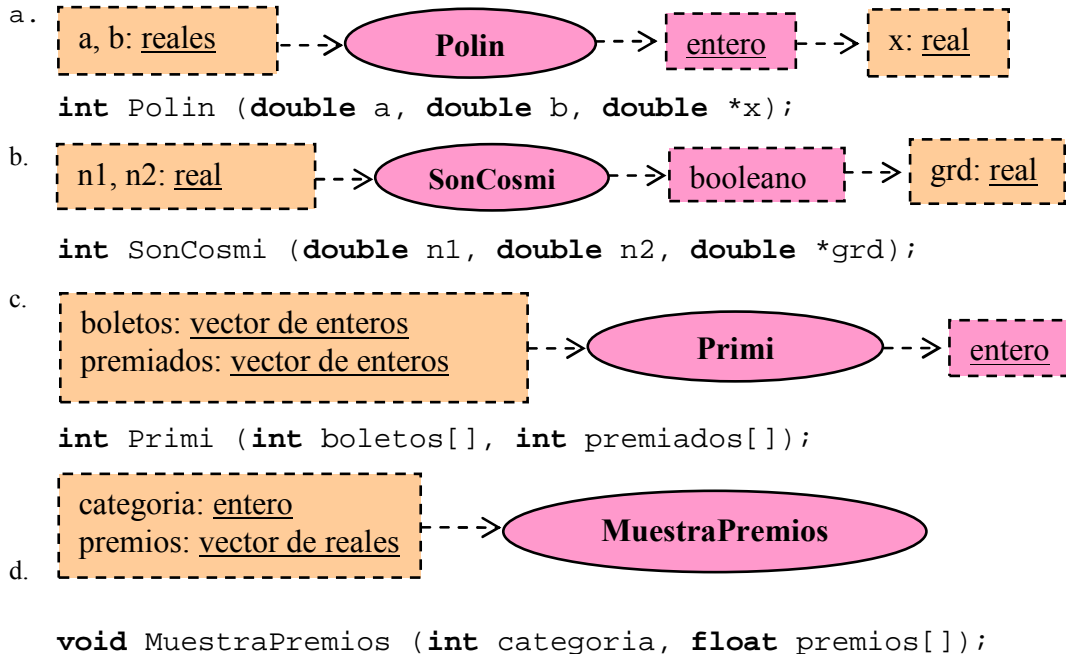
rwX
---
r-x
r--

2.

Sin funciones	Con funciones
<pre>#include &lt;stdio.h&gt;  void main (void) {     int i, j, n;      printf ("Introduce el orden: ");     scanf ("%d", &amp;n);     for (i = 1; i &lt;= n; i++) {         printf ("%d", i);         for (j = 0; j &lt; n - i + 1; j++)             printf ("-");         for (j = 0; j &lt; 2 * i - 1; j++)             printf ("%d", 2 * i - 1);         for (j = 0; j &lt; n - i + 1; j++)             printf ("-");         printf ("\n");     }     for (i = n-1; i &gt; 0; i--) {         printf ("%d", i);         for (j = 0; j &lt; n - i + 1; j++)             printf ("-");         for (j = 0; j &lt; 2 * i - 1; j++)             printf ("%d", 2 * i - 1);         for (j = 0; j &lt; n - i + 1; j++)             printf ("-");         printf ("\n");     } }</pre>	<pre>#include &lt;stdio.h&gt;  void linea (int i, int n);  void main (void) {     int i, n;      printf ("Introduce el orden: ");     scanf ("%d", &amp;n);     for (i = 1; i &lt;= n; i++)         linea (i, n);     for (i = n-1; i &gt; 0; i--)         linea (i, n); }  void linea (int i, int n) {     int j, m;      m = 2 * i - 1;     printf ("%d", i);     for (j = i; j &lt; n; j++) printf ("-");     for (j = 0; j &lt; m; j++) printf ("%d", m);     for (j = i; j &lt; n; j++) printf ("-");     printf ("\n"); }</pre>



3.



4. Diagrama de flujo en la página siguiente.

No sabemos cuántas soluciones hay luego no es adecuado utilizar una estructura **for**.

```
#include <stdio.h>

void main (void)
{
    int fi, n1, n2;
    int i, n;

    printf ("Introduce primer número: ");
    scanf ("%d", &n1);
    printf ("Introduce segundo número: ");
    scanf ("%d", &n2);

    n = 0;
    i = 1;
    do {
        fi = n1 * i + n2 / i;
        if (fi % 2 != 0) {
            n++;
            printf ("%2d i:%2d, fi: %d\n", n, i, fi);
        }
        i++;
    } while (n < 10 && i <= 100);
}
```



Escuela Universitaria  
de Ingeniería  
Vitoria-Gasteiz

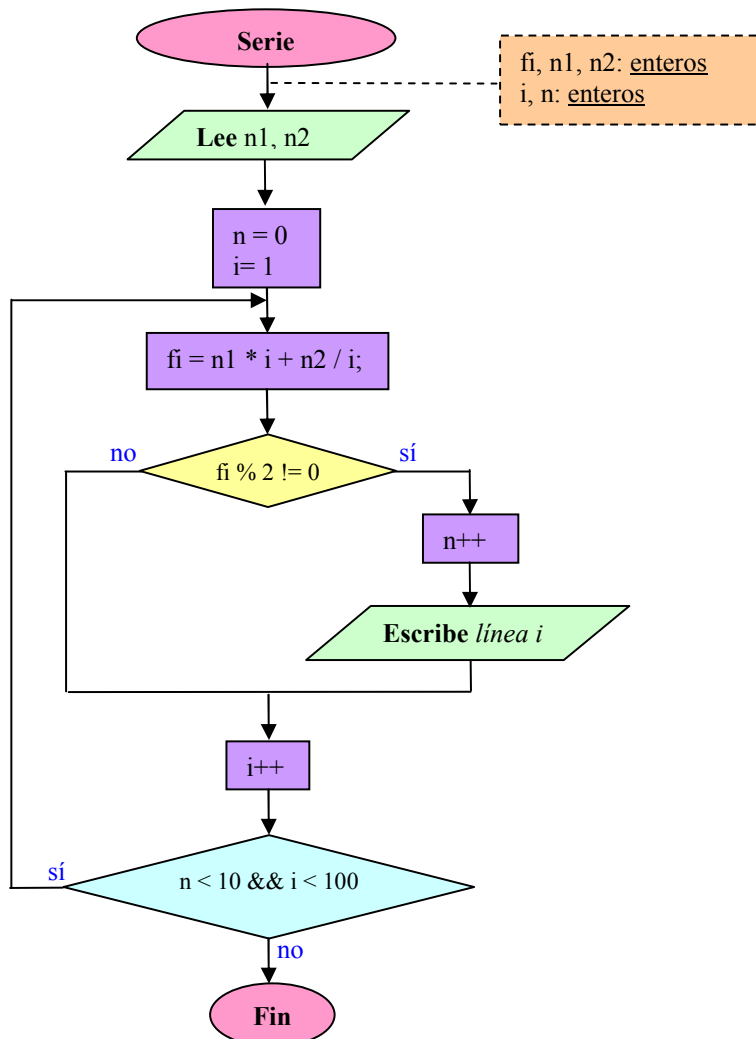
Ingeniaritzako  
Unibertsitate Eskola  
Vitoria-Gasteiz

## Fundamentos de Informática

Examen – 25 de Enero de 2008

**Electrónica**

**Resolución**



5. Para la comprobación se puede utilizar, por ejemplo, una función que asigne ocho horas al día de lunes a jueves, cinco horas y media los viernes y cero horas el fin de semana (esta función no es parte de la solución propuesta):

```
float HorLib (long jul)
{
    int dsm;

    dsm = DiaSem (jul);
    if (dsm >= 0 && dsm <= 3)
        return 8;
    if (dsm == 4)
        return 5.5;
    return 0.0;
}
```



```
#include <stdio.h>
#include "fechas.h"
#include "agenda.h"

void main (void)
{
    int d, m, a;
    long jul;
    float hor, hor1;

    printf ("Introduce horas del proyecto: ");
    scanf ("%f", &hor);
    DiaSis (&d, &m, &a);
    jul = DiaJul (d, m, a);
    while (hor > 0) {
        jul++;
        hor1 = HorLib (jul);
        if (hor1 > 0) {
            DiaGrg (jul, &d, &m, &a);
            printf ("%02d/%02d/%d - ", d, m, a);
            if (hor > hor1) {
                hor -= hor1;
                printf (".2f horas (quedan %6.2f)\n", hor1, hor);
            }
            else {
                printf (".2f horas (finalizado)\n", hor);
                hor = 0;
            }
        }
    }
}
```

6. Empezaremos por proponer un programa de prueba (no es parte de la solución pedida):

```
#include <stdio.h>
#include <math.h>

void bLee (double *c);
void bEsc (double *c);
double bMod (double *c);
void bCjg (double *c);

void main (void)
{
    double c[2];
    bLee (c);
    bEsc (c);
    printf ("Modulo: %.2lf\n", bMod (c));
    bCjg (c);
    bEsc (c);
}
```



Escuela Universitaria  
de Ingeniería  
Vitoria-Gasteiz

Ingeniaritzako  
Unibertsitate Eskola  
Vitoria-Gasteiz

# Fundamentos de Informática

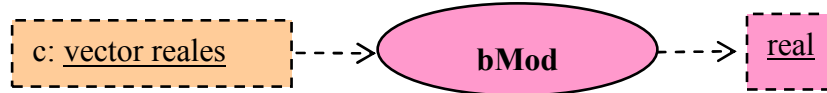
Examen – 25 de Enero de 2008

**Electrónica**

**Resolución**

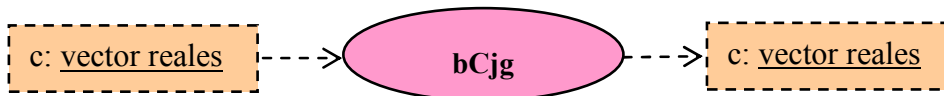


a.



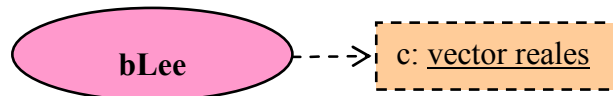
```
double bMod (double *c)
{
    return sqrt (c[0]*c[0] + c[1]*c[1]);
}
```

b.



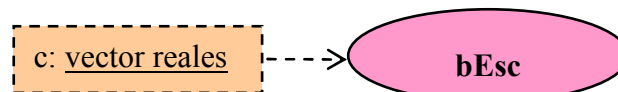
```
void bCjg (double *c)
{
    c[1] = -c[1];
}
```

c.



```
void bLee (double *c)
{
    printf ("Introduce la parte real: ");
    scanf ("%lf", &c[0]);
    printf ("Introduce la parte imaginaria: ");
    scanf ("%lf", &c[1]);
}
```

d.



```
void bEsc (double *c)
{
    if (c[1] >= 0) printf (".2lf + .2lfi", c[0], c[1]);
    else          printf (".2lf - .2lfi", c[0], -c[1]);
}
```