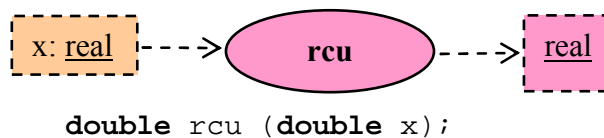


1. [1 punto]

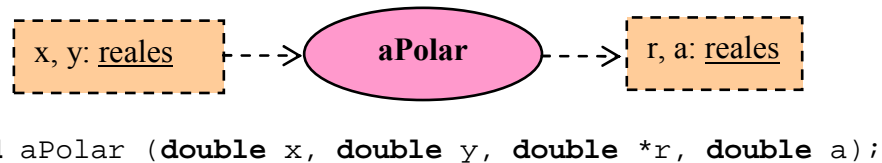
Orden	Expresión	Valor	a	b	c	d	e
0		–	0	1	3	–1	?
1	e = 1.4 + 1.6	3					3
2	e = a?b:c	3					3
3	e = a++	0	1				0
4	e = --a	0	0				0
5	e = b^c	2					2
6	a++ && b	0	1				
7	011 - 10	-1					
8	~0xa	-11					
9	e=c<<2	12					12
10	e= ~d	0					0

2. [1 punto]

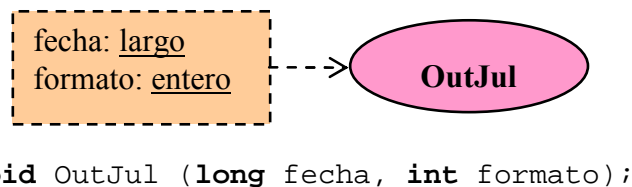
a. Raíz cuadrada.



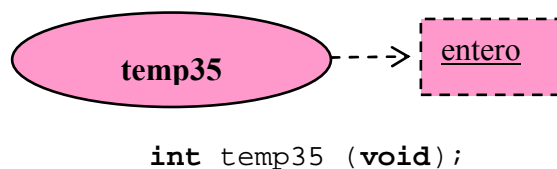
b. A polar.



c. Muestra fecha.



d. Lee temperatura.





3. [3 puntos]

- a. Primeros 15 elementos pares de la serie. (1,75 puntos)

```
#include <stdio.h>

void main(void)
{
    int i, n;
    int x;

    n = 0;
    i = 1;
    do {
        x = S (i);
        if (x % 2 == 0) {
            printf ("%d ", x);
            n++;
        }
        i++;
    } while (n < 15);
}
```

- b. Índice de x. (1,25 puntos)

```
int IndEnS(int x)
{
    int i;
    i = 1;
    while (S(i) < x)
        i++;
    if (S(i) == x) return i;
    else          return 0;
}
```

Alternativa:

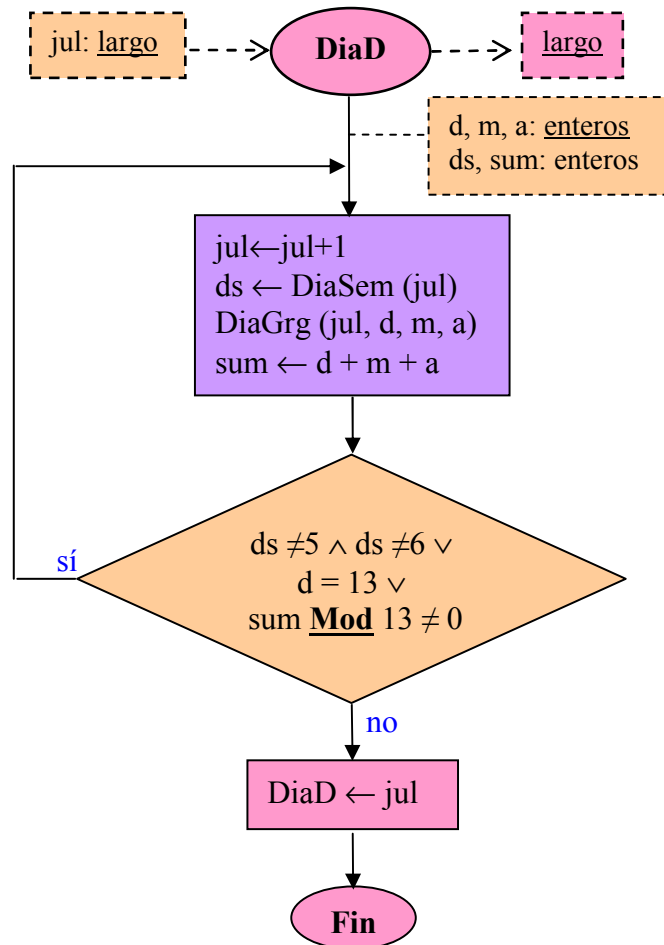
```
int IndEnS(int x)
{
    int i;
    i = 1;
    while (S(i) < x)
        i++;
    return S(i)==x? i:0;
}
```

Alternativa (válida pero no recomendable ya que no es un bucle **for** natural):

```
int IndEnS(int x)
{
    int i;
    for (i = 1; S(i) < x; i++);
    return S(i)==x? i:0;
}
```

4. [2,5 puntos] Siguiente fecha especial.

a) **Diseño** siguiente fecha especial: (1,50 puntos)



Código siguiente fecha especial:

```

long DiaD (long jul)
{
    int ds, sum;
    int d, m, a;
    do {
        jul++;
        ds = DiaSem (jul);
        DiaGrg (jul, &d, &m, &a);
        sum = d + m + a;
    } while (ds != 5 && ds != 6 || d == 13 || sum % 13 != 0);
    return jul;
}
    
```

b) **Código** función C (programa principal) (1 punto):

```

void main(void)
{
    long hoy, sig;
    int d, m, a;
    DiaSis (&d, &m, &a);
    hoy = DiaJul (d, m, a);
    sig = DiaD(hoy);
    DiaGrg (sig, &d, &m, &a);
    printf ("La siguiente: %02d/%02d/%d\n", d, m, a);
}
    
```



5. [2,5 puntos] Código función C:

```
int proposicion(char c1, char c2, char *frase)
{
    int i, k;

    i = 0;
    while (frase[i] != c1 && frase[i] != '\0')
        i++;

    k = 0;
    while (k < 3 && frase[i] != '\0') {
        if (frase[i] == c2)
            k++;
        i++;
    }
    return k == 3; /* Si cierto devuelve 1, si no 0 */
}
```