

Módulo I: Algoritmos y Estructura de Datos

Alumno: Gorosito, Matías

Legajo: 140934-7

Profesor: Pablo A. Sznajdleder

Materia: Algoritmos y Estructura de Datos

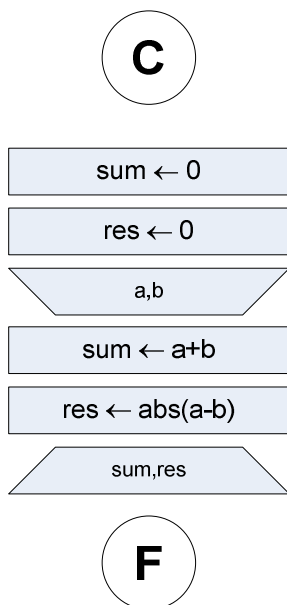
Año: 2010

Curso: K1124

Ejercicio N° 1

A partir de dos valores enteros A y B, informar la suma, la diferencia entre A y B, y el producto.

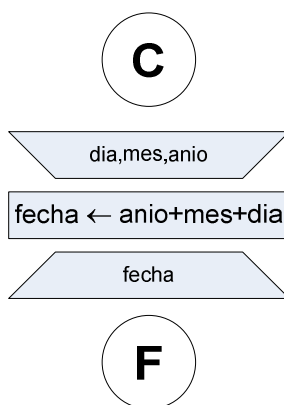
```
var a,b,sum,res,prod:integer;  
begin  
    writeln('Ingrese un numero:');  
    readln(a);  
    writeln('Ingrese un numero:');  
    readln(b);  
    sum:=a+b;  
    res:=abs(a-b);  
    prod:=a*b;  
    writeln('La suma es:', sum);  
    writeln('La diferencia es:', res);  
    writeln('El producto es:', prod);  
end.
```



Ejercicio N° 3

Dada una terna de números naturales que representan al día, al mes y al año de una determinada fecha informarla como un solo número natural de 8 dígitos (AAAAMMDD).

```
var dia,mes:string[2];
var anio:string[4];
var fecha:string[8];
begin
    writeln('Ingrese el dia:');
    readln(dia);
    writeln('Ingrese el mes:');
    readln(mes);
    writeln('Ingrese el anio:');
    readln(anio);
    fecha:= anio+mes+dia;
    writeln('La fecha ingresada es:', fecha);
end.
```



Ejercicio N° 5

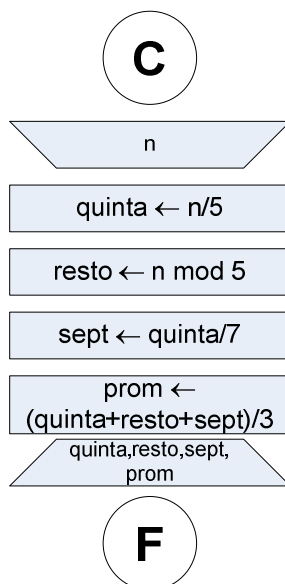
A partir de un valor entero ingresado por teclado, se pide informar:

1. La quinta parte de dicho valor
2. El resto de la división por 5
3. La séptima parte del resultado del punto 1
4. El promedio real de los valores obtenidos en 1, 2, y 3

```

var quinta,sept:double;
var prom:real;
var resto,n:integer;
begin
    writeln('Ingrese un numero');
    readln(n);
    quinta:=(n/5);
    resto:= n mod 5;
    sept:=(quinta/7);
    prom:=((quinta+resto+sept)/3);
    writeln('La quinta parte de:',quinta);
    writeln('El resto de la division por 5:',resto);
    writeln('La septima parte de la quinta parte de:',sept);
    writeln('El promedio real de los valores obtenidos en
1,2,3:',prom);
end.

```

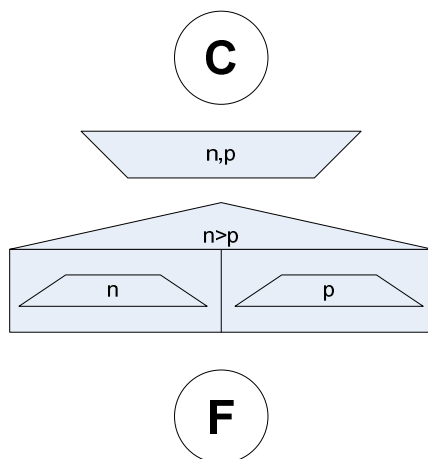


Ejercicio N° 6

Dados dos valores enteros y distintos, emitir una leyenda apropiada que informe cuál

es el mayor entre ellos.

```
var n,p:integer;
begin
  writeln('Ingrese un numero:');
  readln(n);
  writeln('Ingrese otro numero:');
  readln(p);
  while (n=p) do begin
    writeln('Ha ingresado un numero igual al primero que
se le pidio, por favor ingrese uno distinto:');
    readln(p);
  end;
  if (n>p) then begin
    writeln('El numero ',n,' es el mayor');
  end else begin
    writeln('El numero ',p,' es el mayor');
  end;
end.
```



Ejercicio N°8

Dadas dos fechas informar cual es la más reciente. Determine cuales serían los datos de entrada y las leyendas a informar de acuerdo al proceso solicitado.

```
function getAnio(fec:longint):integer;
begin
  getAnio:=fec div 10000;
end;
function getMes(fec:longint):integer;
```

```

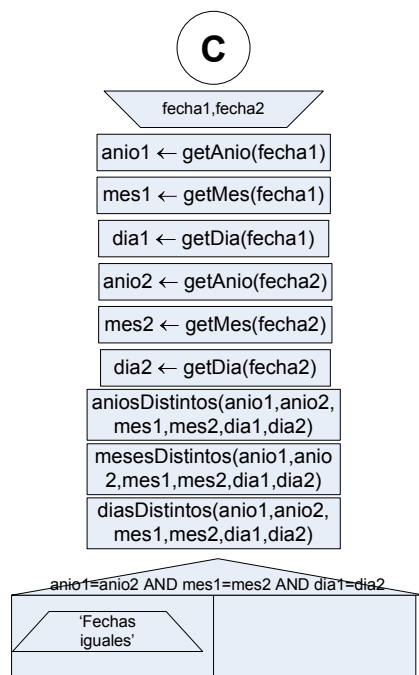
var x:integer;
begin
    x:=fec mod 10000;
    getMes:=x div 100;
end;
function getDia(fec:longint):integer;
begin
    getDia:=fec mod 100;
end;
procedure
aniosDistintos(anio1,anio2,mes1,mes2,dia1,dia2:integer);
begin
    if (anio1>anio2) then begin
        writeln(dia1,'/',mes1,'/',anio1,' es la fecha
ingresada mas reciente');
    end;
    if (anio1<anio2) then begin
        writeln(dia2,'/',mes2,'/',anio2,' es la fecha
ingresada mas reciente');
    end;
end;
procedure
mesesDistintos(anio1,anio2,mes1,mes2,dia1,dia2:integer);
begin
    if (anio1=anio2) then begin
        if (mes1>mes2) then begin
            writeln(dia1,'/',mes1,'/',anio1,' es la fecha
ingresada mas reciente');
        end;
        if (mes1<mes2) then begin
            writeln(dia2,'/',mes2,'/',anio2,' es la fecha
ingresada mas reciente');
        end;
    end;
end;
procedure
diasDistintos(anio1,anio2,mes1,mes2,dia1,dia2:integer);
begin
    if (anio1=anio2) and (mes1=mes2) then begin

```

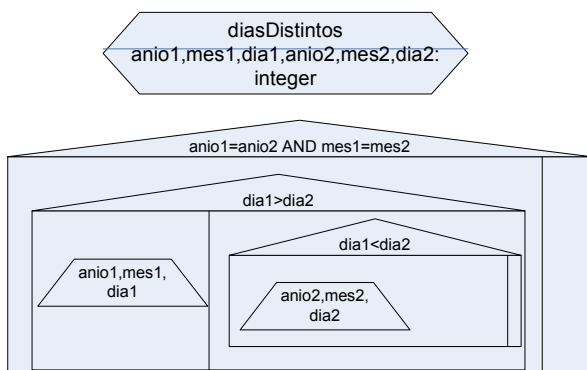
```

        if (dia1>dia2) then begin
            writeln(dia1,'/',mes1,'/',anio1,' es la fecha
ingresada mas reciente');
        end;
        if (dia1<dia2) then begin
            writeln(dia2,'/',mes2,'/',anio2,' es la fecha
ingresada mas reciente');
        end;
    end;
end;
var fecha:longint;
var anio1,mes1,dia1,anio2,mes2,dia2:integer;
begin
    writeln('Ingresa una fecha en formato (AAAAMMDD):');
    readln(fecha);
    anio1:=getAnio(fecha);
    mes1:=getMes(fecha);
    dia1:=getDia(fecha);
    writeln('Ingresa otra fecha en formato (AAAAMMDD):');
    readln(fecha);
    anio2:=getAnio(fecha);
    mes2:=getMes(fecha);
    dia2:=getDia(fecha);
    aniosDistintos(anio1,anio2,mes1,mes2,dia1,dia2);
    mesesDistintos(anio1,anio2,mes1,mes2,dia1,dia2);
    diasDistintos(anio1,anio2,mes1,mes2,dia1,dia2);
    if (anio1=anio2) and (mes1=mes2) and (dia1=dia2) then
begin
        writeln('Las dos fechas son iguales');
    end;
end.

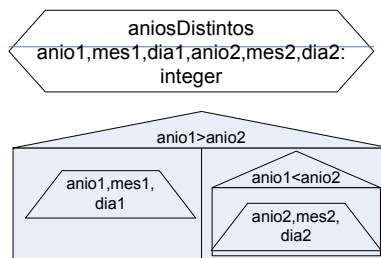
```



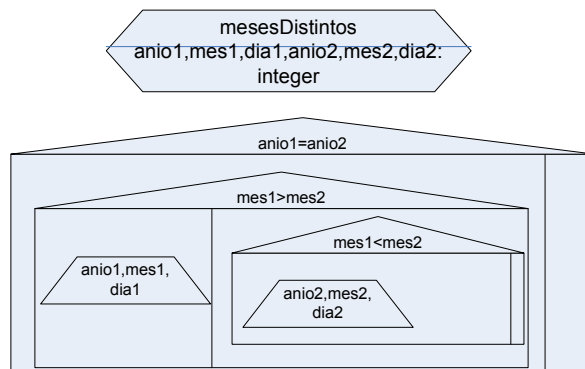
F



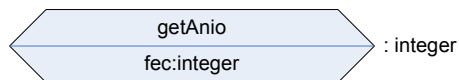
R



R

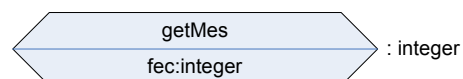


R



getAnio ← fec div 10000

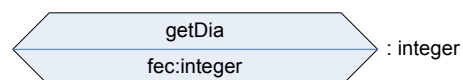
R



x ← fec mod 10000

getMes ← x div 100

R



getDia ← fec mod 100

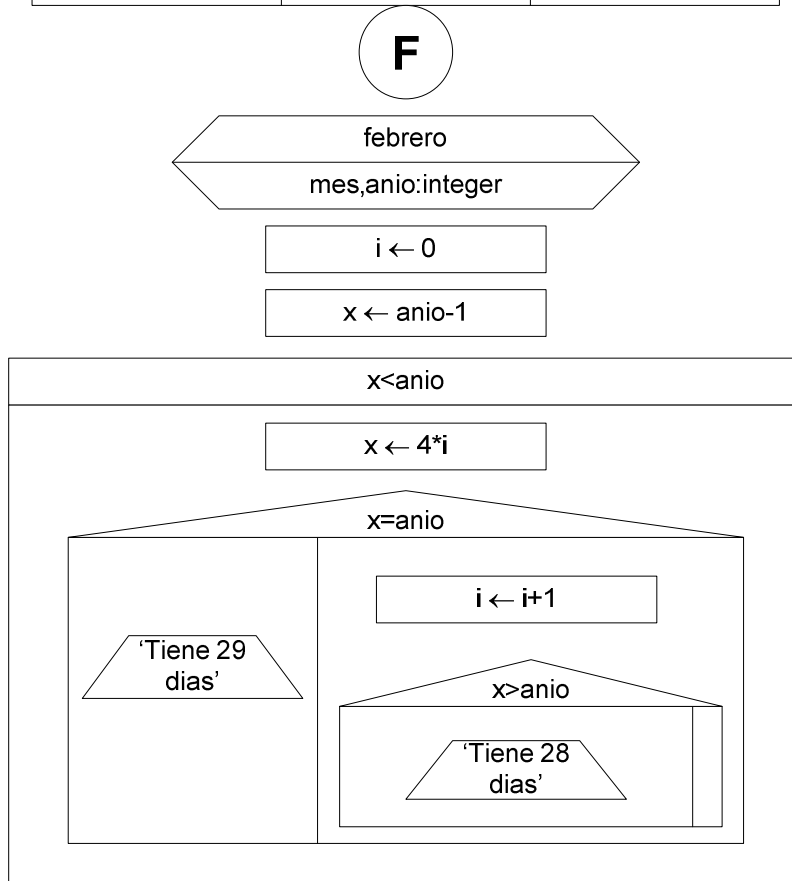
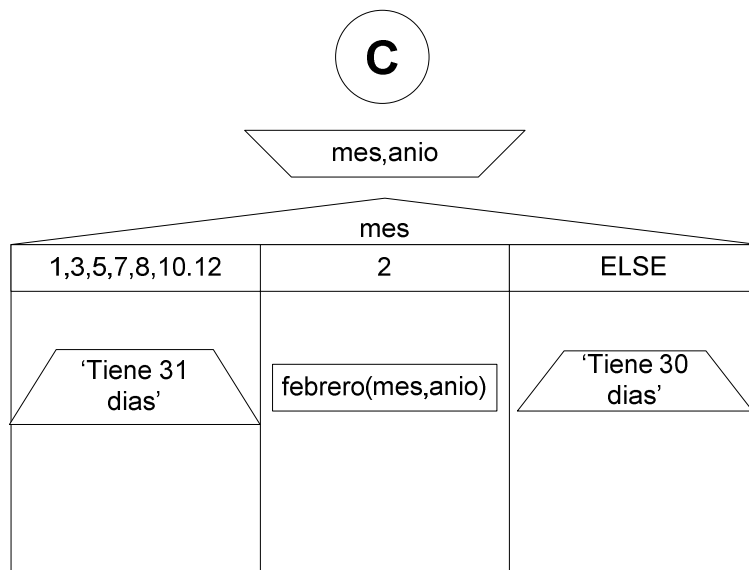
R

Ejercicio N° 10

Dados un mes y el año correspondiente informar cuantos días tiene el mes.

```
procedure febrero(mes,anio:integer);
var i,x:integer;
begin
    i:=0;
    x:=anio-1;
    while (x<anio) do begin
        x:=4*i;
        if (x=anio) then begin
            writeln('Tiene 29 dias');
        end else begin
            i:=i+1;
            if (x>anio) then begin
                writeln('Tiene 28 dias');
            end;
        end;
    end;
end;

var mes,anio:integer;
begin
    writeln('Ingrese un mes (en formato MM):');
    readln(mes);
    writeln('Ingrese un anio (en forma AAAA):');
    readln(anio);
    case mes of
        1,3,5,7,8,10,12: begin writeln('Tiene 31 dias');end;
        2: begin febrero(mes,anio); end;
        else begin writeln('Tiene 30 dias');end;
    end;
end.
```



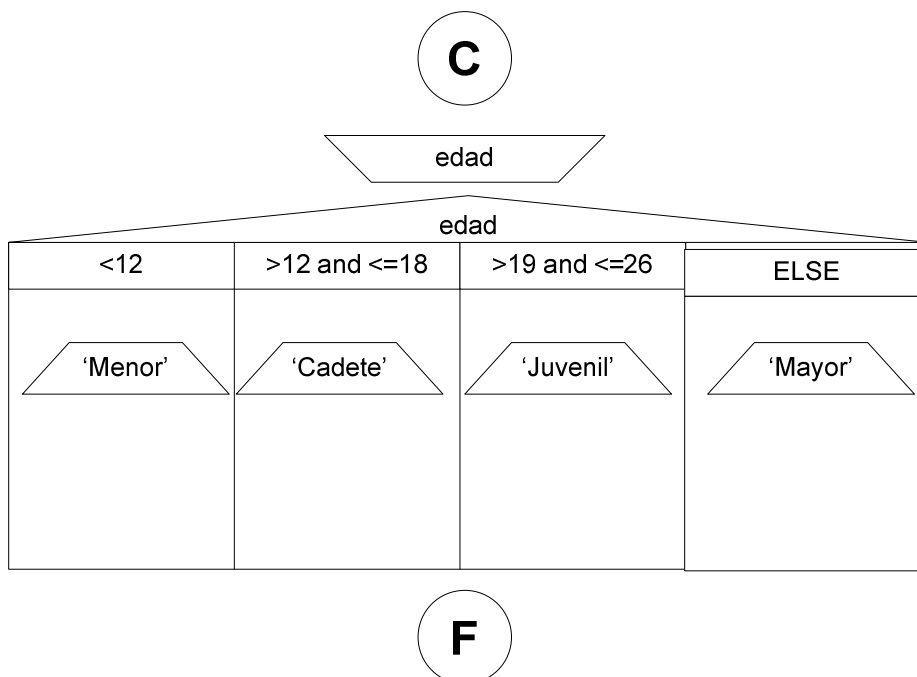
R

Ejercicio N° 12

Se ingresa una edad, mostrar por pantalla alguna de las siguientes leyendas:

- 'menor' si la edad es menor o igual a 12
- 'cadete' si la edad esta comprendida entre 13 y 18
- 'juvenil' si la edad es mayor que 18 y no supera los 26
- 'mayor' en el caso que no cumpla ninguna de las condiciones anteriores

```
var edad:integer;  
begin  
  writeln('Ingrese una edad:');  
  readln(edad);  
  case edad of  
    1..12 : begin writeln('Menor');end;  
    13..18 : begin writeln('Cadete'); end;  
    19..26 : begin writeln('Juvenil');end;  
    else begin writeln('Mayor'); end;  
  end;  
end.
```



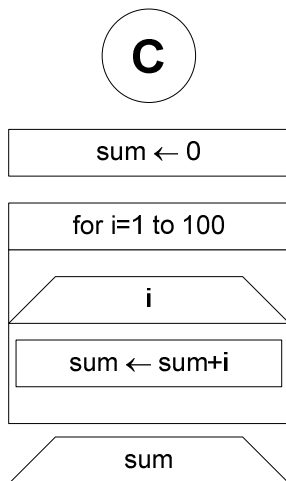
Ejercicio N° 13

Informar los primeros 100 números naturales y su sumatoria.

```

var i,sum:integer;
begin
    sum:=0;
    for i:=1 to 100 do begin
        writeln(i);
        sum:=sum+i;
    end;
    writeln('La sumatoria de los primeros 100 numeros
naturales es: ', sum);
end.

```



F

Ejercicio N° 15

Dados N y M números naturales, informar su producto por sumas sucesivas.

```

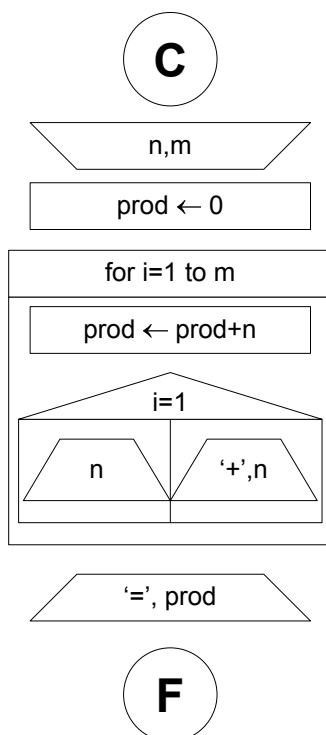
var n,m:word;
var prod:longint;
var i:integer;
begin
    writeln('Ingrese un numero natural:');
    readln(n);
    writeln('Ingrese otro numero natural:');
    readln(m);
    prod:= 0;
    writeln('Calculando el producto...');

```

```

    for i:=1 to m do begin
        prod:= prod + n;
        if (i=1) then begin
            write(n);
        end else begin
            write('+',n);
        end;
    end;
    writeln('=',prod);
end.

```



Ejercicio N° 16

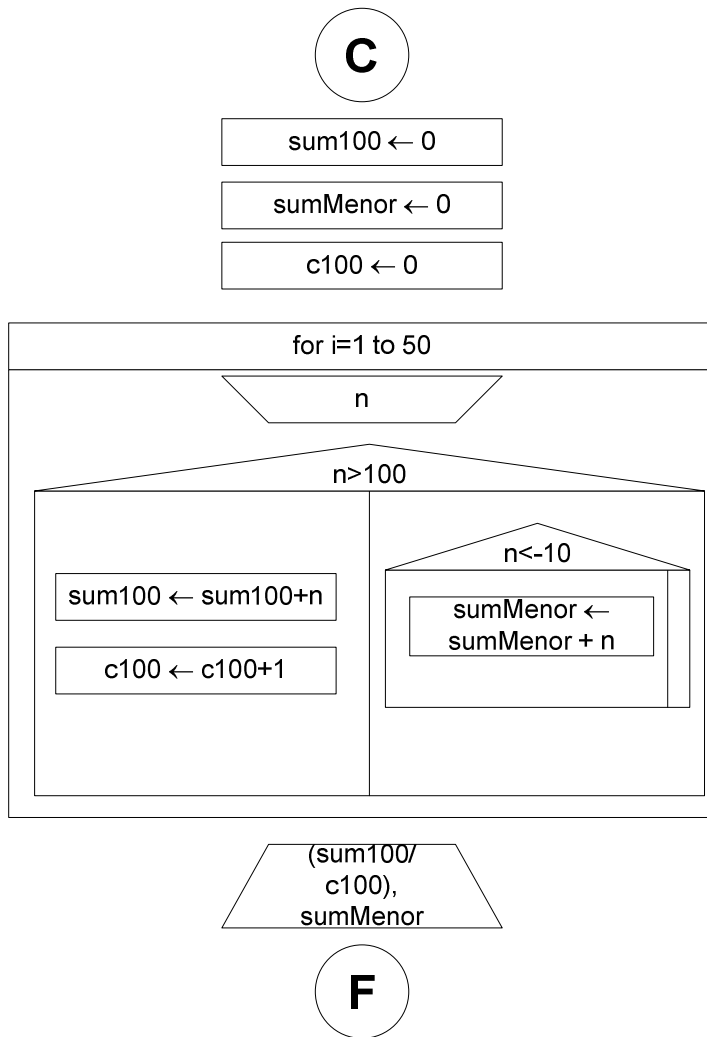
Dados 50 números enteros, informar el promedio de los mayores a 100 y la suma de los menores que -10.

```

var n,i,sum100,c100,summenor:integer;
var prom:single;
begin
    writeln('Ingrese un numero:');
    readln(n);
    i:=1;
    c100:=0;

```

```
sum100:=0;
summenor:=0;
while (i<50) do begin
    if (n>100) then begin
        sum100:=sum100+n;
        c100:=c100+1;
    end;
    if (n<-10) then begin
        summenor:=summenor+n;
    end;
    i:=i+1;
    writeln('Ingrese otro numero:');
    readln(n);
end;
prom:=sum100/c100;
writeln('Promedio de los mayores a 100:', prom);
writeln('Suma de los menores a -10: ', summenor);
end.
```



Ejercicio N° 17

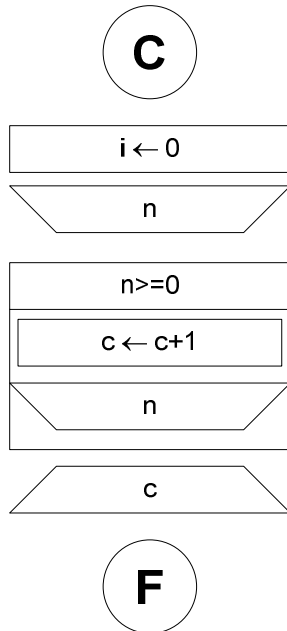
Ingresar e informar valores, mientras que el valor ingresado no sea negativo.
Informar la cantidad de valores ingresados.

```

var n,c:integer;
begin
  writeln('Ingrese un numero:');
  readln(n);
  c:=0;
  while (n >= 0) do begin
    c:=c+1;
    writeln('Ingrese otro numero:');
    readln(n);
  end;

```

```
writeln('Cantidad de valores ingresados: ', c);
end.
```



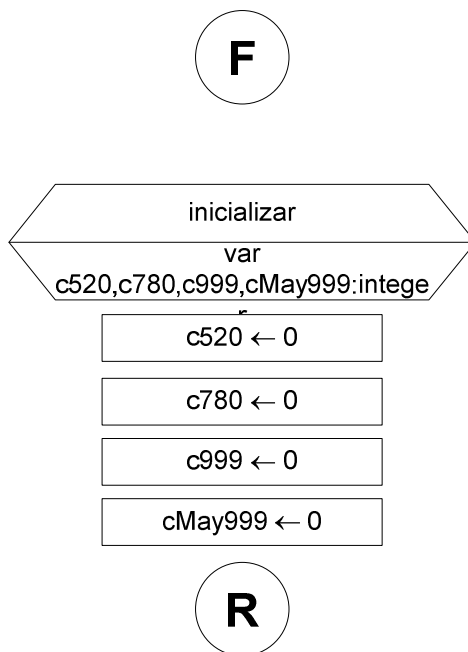
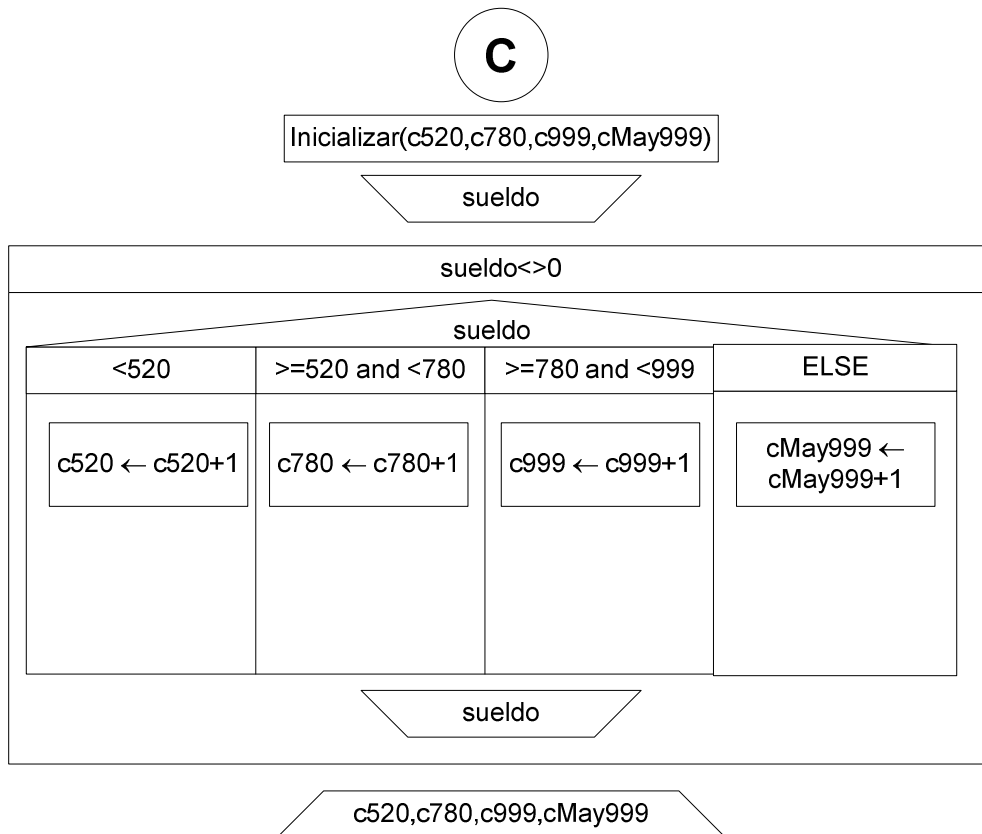
Ejercicio N° 19

Se ingresa un conjunto de valores reales, cada uno de los cuales representan el sueldo de un empleado, excepto el último valor que es cero e indica el fin del conjunto. Se pide desarrollar un programa que determine e informe:

1. Cuántos empleados ganan menos \$520.
2. Cuántos ganan \$520 o más pero menos de \$780.
3. Cuántos ganan \$780 o más pero menos de \$999.
4. Cuántos ganan \$999 o más.

```
var sueldo:integer;
var c520,c780,c999,cmay999:integer;
begin
  writeln('Ingrese el sueldo a calcular:');
  readln(sueldo);
  c520:=0;
  c780:=0;
  c999:=0;
  cmay999:=0;
  while (sueldo <>0) do begin
    case sueldo of
```

```
        0 .. 519 : begin c520:=c520+1; end;
        520 .. 779 : begin c780:=c780+1; end;
        780 .. 998 : begin c999:=c999+1; end;
        else begin cmay999:=cmay999+1; end;
    end;
    writeln('Ingrese otro sueldo:');
    readln(sueldo);
end;
    writeln('Cantidad de empleados que ganan menos de $520:
',c520);
    writeln('Cantidad de empleados que ganan $520 o mas pero
menos de $780: ',c780);
    writeln('Cantidad de empleados que ganan $780 o mas pero
menos de $999: ',c999);
    writeln('Cantidad de empleados que ganan $999 o mas:
',cmay999);
end.
```



Ejercicio N° 21

Dado un conjunto de sueldos a pagar (sin centavos) que finaliza con un valor cero, desarrollar un programa que determine e informe cuantos billetes de \$100, \$50, \$20,

\$10, \$5, \$2 y monedas de \$1 son necesarios para ensobrar los sueldos y no tener dificultades con el cambio (utilizando la mínima cantidad de billetes).

```
procedure mostrarBilletes(b100,b50,b20,b10,b5,b2,m1:integer);
begin
    writeln('Cantidad de billetes de $100: ',b100);
    writeln('Cantidad de billetes de $50: ',b50);
    writeln('Cantidad de billetes de $20: ',b20);
    writeln('Cantidad de billetes de $10: ',b10);
    writeln('Cantidad de billetes de $5: ',b5);
    writeln('Cantidad de billetes de $2: ',b2);
    writeln('Cantidad de monedas de $1: ',m1);
end;

procedure inicBilletes(var b100,b50,b20,b10,b5,b2,m1,n:integer);
begin
    b100:=0;
    b50:=0;
    b20:=0;
    b10:=0;
    b5:=0;
    b2:=0;
    m1:=0;
    n:=100;
end;

procedure contBilletes(n:integer; var
b100,b50,b20,b10,b5,b2,m1:integer);
begin
    case n of
        100 : begin b100:=b100+1; end;
        50  : begin b50:=b50+1; end;
        20  : begin b20:=b20+1; end;
        10  : begin b10:=b10+1; end;
        5   : begin b5:=b5+1; end;
        2   : begin b2:=b2+1; end;
        1   : begin m1:=m1+1; end;
    end;
end;
```

```

procedure camBilletes(var n:integer);
begin
    case n of
        100 : begin n:=50; end;
        50  : begin n:=20; end;
        20  : begin n:=10; end;
        10  : begin n:=5;  end;
        5   : begin n:=2;  end;
        2   : begin n:=1;  end;
    end;
end;

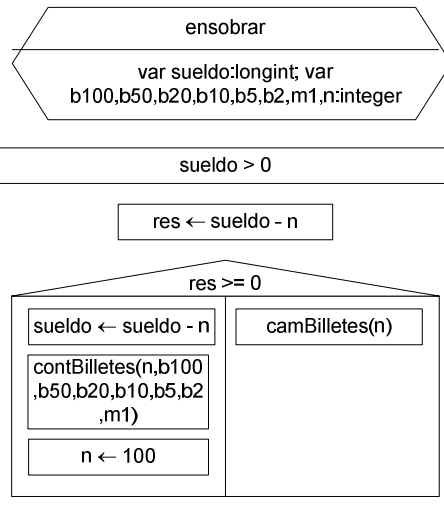
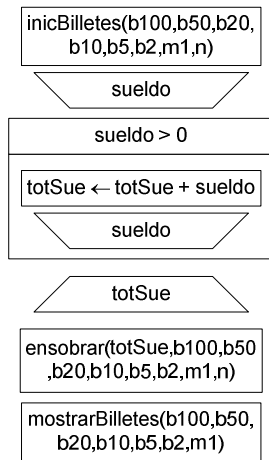
procedure ensobrar(var sueldo:longint; var
b100,b50,b20,b10,b5,b2,m1,n:integer);
var res:integer;
begin
    while (sueldo>0) do begin
        res:=sueldo-n;
        if (res>=0) then begin
            sueldo:=sueldo-n;
            contBilletes(n,b100,b50,b20,b10,b5,b2,m1);
            n:=100;
        end else begin
            camBilletes(n);
        end;
    end;
end;

var sueldo,totSue:longint;b100,b50,b20,b10,b5,b2,m1,n:integer;
begin
    writeln('Ingrese el sueldo a calcular:');
    inicBilletes(b100,b50,b20,b10,b5,b2,m1,n);
    readln(sueldo);
    while (sueldo<> 0) do begin
        totSue:=totSue+sueldo;
        writeln('Ingrese otro sueldo:');
        readln(sueldo);
    end;
end;

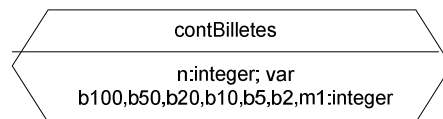
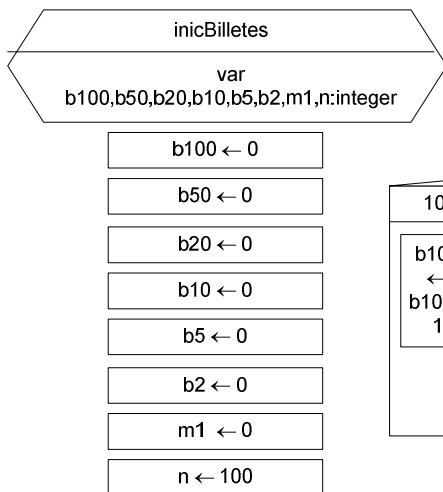
```

```
writeln('Total de Sueldos: ', totSue);  
ensobrar(totSue, b100, b50, b20, b10, b5, b2, m1, n);  
mostrarBilletes(b100, b50, b20, b10, b5, b2, m1);  
end.
```

C



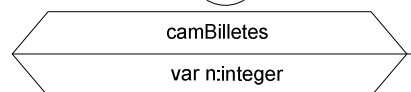
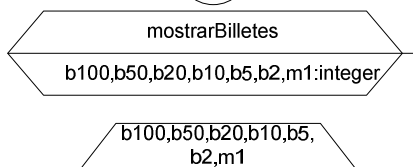
F



n						
100	50	20	10	5	2	1
b100 ← b100+ 1	b50 ← b50+1	b20 ← b20+1	b10 ← b10+1	b5 ← b5+1	b2 ← b2+1	m1 ← m1+1

R

R



n					
100	50	20	10	5	2
n ← 50	n ← 20	n ← 10	n ← 5	n ← 2	n ← 1

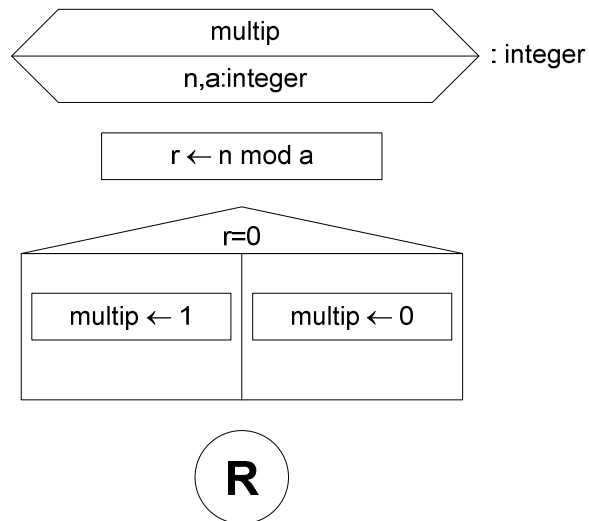
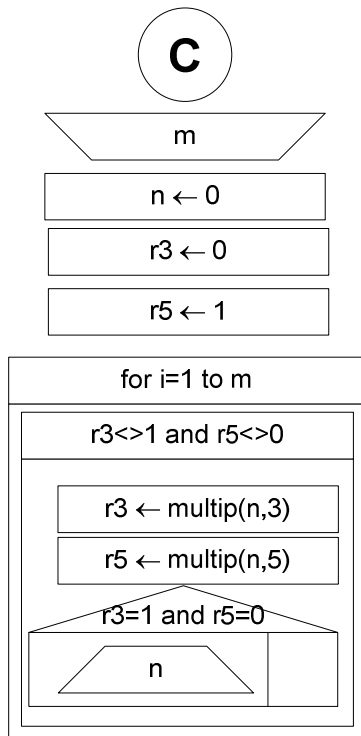
R

Ejercicio N° 22

Dado un valor M determinar y emitir un listado con los M primeros múltiplos de 3 que no lo sean de 5, dentro del conjunto de los números naturales.

```
function multip(n,a:integer):integer;
var r:integer;
begin
    r:= n mod a;
    if (r=0) then begin
        multip:=1;
    end else begin
        multip:=0;
    end;
end;

var m,n,r3,r5,i:integer;
begin
    writeln('Ingrese un numero:');
    read(m);
    n:=0;
    r3:=0;
    r5:=1;
    writeln('Estos son los ',m,' primeros multiplos de 3 que
no lo son de 5');
    for i:=1 to m do begin
        while ((r3<>1) and (r5<>0)) do begin
            r3:=multip(n,3);
            r5:=multip(n,5);
            if ((r3=1) and (r5=0)) then begin
                writeln(n);
            end;
            n:=n+1;
        end;
    end;
end.
```



F

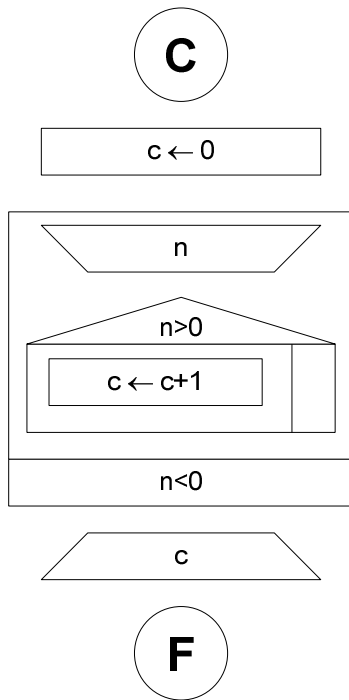
Ejercicio N° 23

Rescribir los algoritmos MI-17 al MI-22, utilizando Repeat-Until.

1-

```

var n,c:integer;
begin
  c:=0
  repeat
    writeln('Ingrese un numero:');
    readln(n);
    if (n>0) then begin
      c:=c+1;
    end;
  until (n<0);
  writeln('Cantidad de valores ingresados: ', c);
end.
  
```



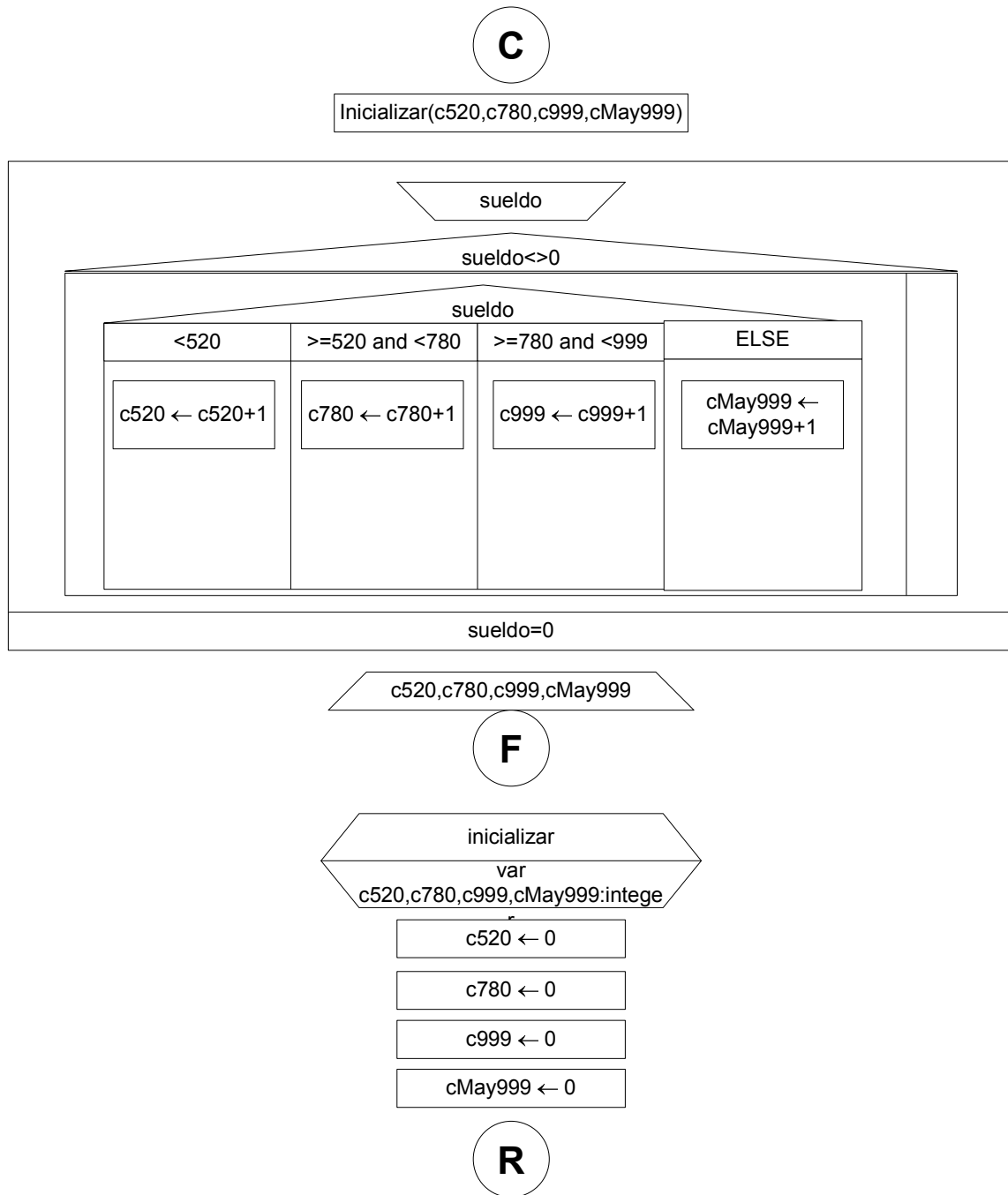
2-

```

var sueldo:integer;
var c520,c780,c999,cmay999:integer;
begin
  c520:=0;
  c780:=0;
  c999:=0;
  cmay999:=0;
  repeat
    writeln('Ingrese el sueldo a calcular:');
    readln(sueldo);
    if (sueldo<>0) then begin
      case sueldo of
        1 .. 519 : begin c520:=c520+1; end;
        520 .. 779 : begin c780:=c780+1; end;
        780 .. 998 : begin c999:=c999+1; end;
        else begin cmay999:=cmay999+1; end;
      end;
    end;
  until (sueldo=0);
  writeln('Cantidad de empleados que ganan menos de $520:
',c520);

```

```
writeln('Cantidad de empleados que ganan $520 o mas pero
menos de $780: ',c780);
writeln('Cantidad de empleados que ganan $780 o mas pero
menos de $999: ',c999);
writeln('Cantidad de empleados que ganan $999 o mas:
',cmay999);
end.
```



3-

```

procedure mostrarBilletes(b100,b50,b20,b10,b5,b2,m1:integer);
begin
  writeln('Cantidad de billetes de $100: ',b100);
  writeln('Cantidad de billetes de $50: ',b50);
  writeln('Cantidad de billetes de $20: ',b20);
  writeln('Cantidad de billetes de $10: ',b10);

```

```

        writeln('Cantidad de billetes de $5: ',b5);
        writeln('Cantidad de billetes de $2: ',b2);
        writeln('Cantidad de monedas de $1: ',m1);
    end;

    procedure inicBilletes(var b100,b50,b20,b10,b5,b2,m1,n:integer);
    begin
        b100:=0;
        b50:=0;
        b20:=0;
        b10:=0;
        b5:=0;
        b2:=0;
        m1:=0;
        n:=100;
    end;

    procedure contBilletes(n:integer; var
    b100,b50,b20,b10,b5,b2,m1:integer);
    begin
        case n of
            100 : begin b100:=b100+1; end;
            50  : begin b50:=b50+1; end;
            20  : begin b20:=b20+1; end;
            10  : begin b10:=b10+1; end;
            5   : begin b5:=b5+1; end;
            2   : begin b2:=b2+1; end;
            1   : begin m1:=m1+1; end;
        end;
    end;

    procedure camBilletes(var n:integer);
    begin
        case n of
            100 : begin n:=50; end;
            50  : begin n:=20; end;
            20  : begin n:=10; end;
            10  : begin n:=5; end;
            5   : begin n:=2; end;
        end;
    end;

```

```

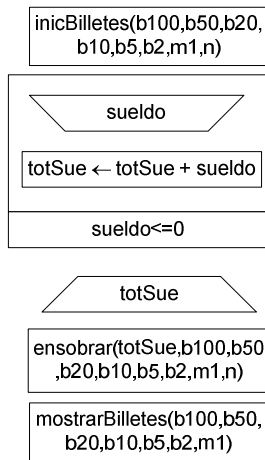
        2 : begin n:=1; end;
    end;
end;

procedure ensobrar(var sueldo:longint; var
b100,b50,b20,b10,b5,b2,m1,n:integer);
var res:integer;
begin
    repeat
        res:=sueldo-n;
        if (res>=0) then begin
            sueldo:=sueldo-n;
            contBilletes(n,b100,b50,b20,b10,b5,b2,m1);
            n:=100;
        end else begin
            camBilletes(n);
        end;
    until (sueldo=0);
end;

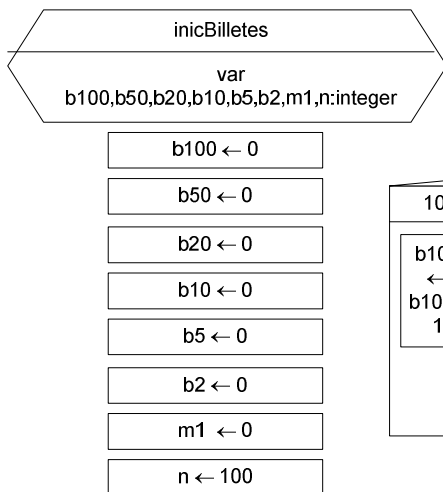
var sueldo,totSue:longint;b100,b50,b20,b10,b5,b2,m1,n:integer;
begin
    inicBilletes(b100,b50,b20,b10,b5,b2,m1,n);
    repeat
        writeln('Ingrese el sueldo a calcular:');
        readln(sueldo);
        totSue:=totSue+sueldo;
    until (sueldo<=0);
    writeln('Total de Sueldos: ',totSue);
    ensobrar(totSue,b100,b50,b20,b10,b5,b2,m1,n);
    mostrarBilletes(b100,b50,b20,b10,b5,b2,m1);
end.

```

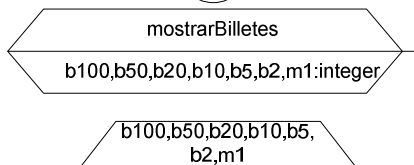
C



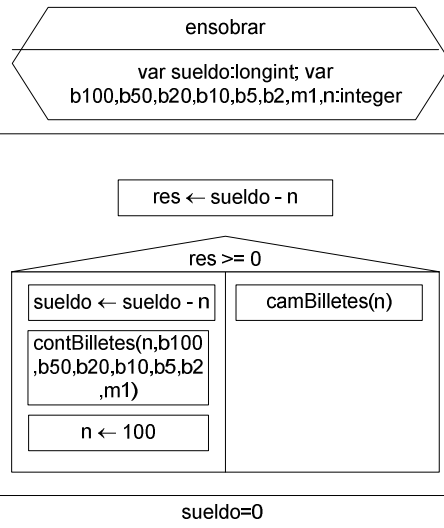
F



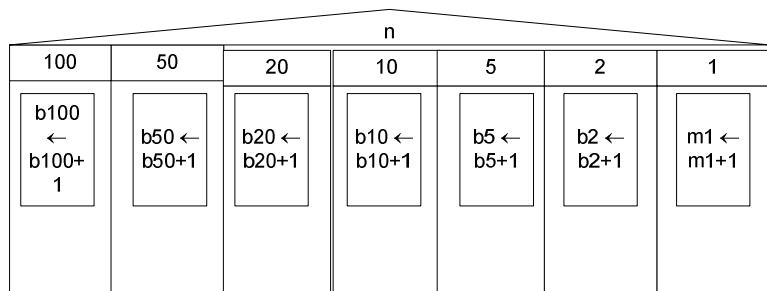
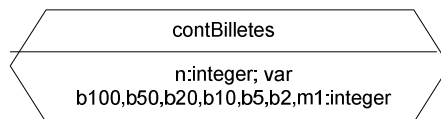
R



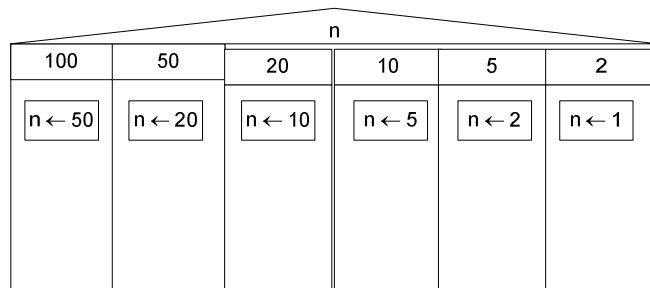
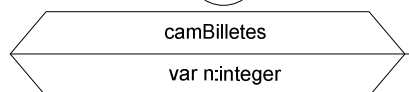
R



R



R



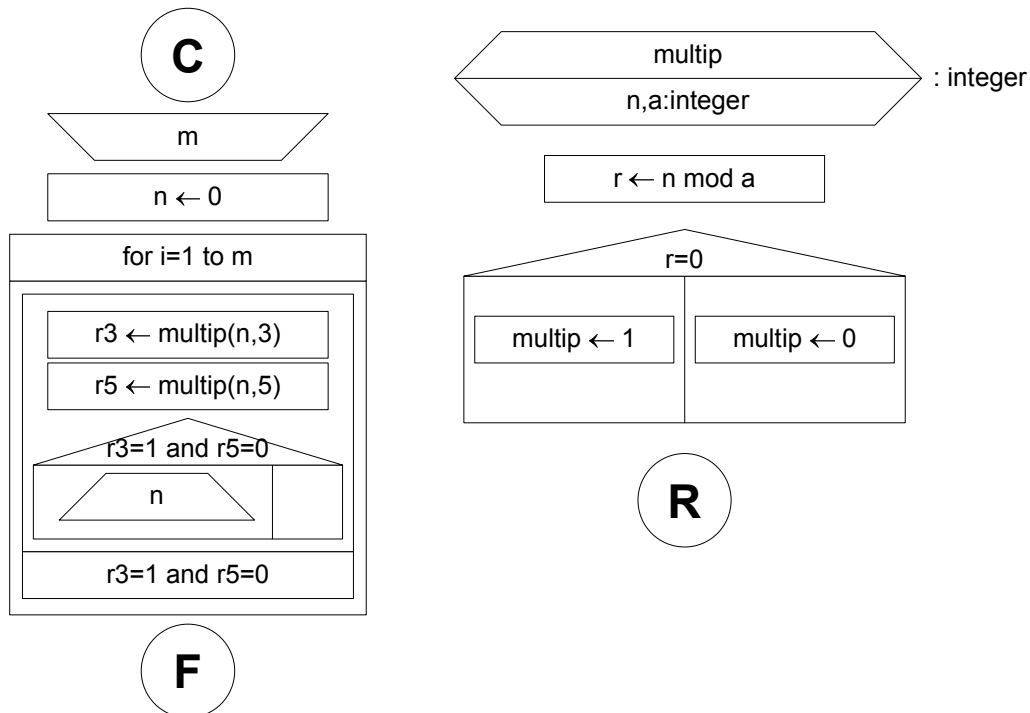
R

```

4-
function multip(n,a:integer):integer;
var r:integer;
begin
    r:= n mod a;
    if (r=0) then begin
        multip:=1;
    end else begin
        multip:=0;
    end;
end;

var m,n,r3,r5,i:integer;
begin
    writeln('Ingrese un numero:');
    read(m);
    n:=0;
    writeln('Estos son los ',m,' primeros multiplos de 3 que
no lo son de 5');
    for i:=1 to m do begin
        repeat
            r3:=multip(n,3);
            r5:=multip(n,5);
            if ((r3=1) and (r5=0)) then begin
                writeln(n);
            end;
            n:=n+1;
        until ((r3=1) and (r5=0));
    end;
end.

```

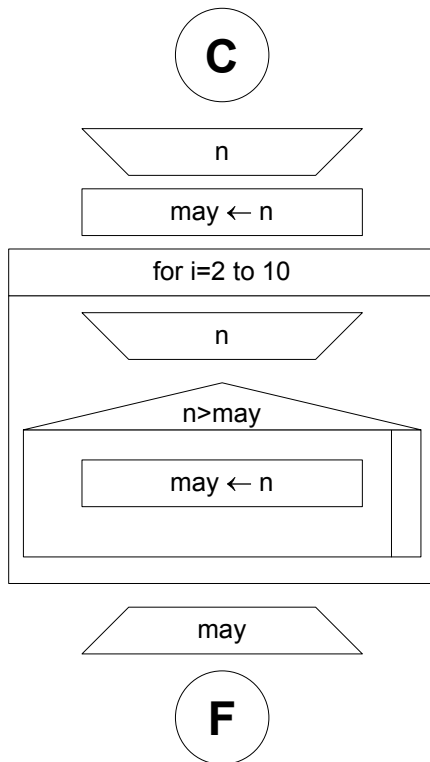


Ejercicio N° 26

Dados 10 valores informar el mayor.

```

var n,may,i:integer;
begin
  writeln('Ingrese un numero: ');
  readln(n);
  may:=n;
  for i:=2 to 10 do begin
    writeln('Ingrese otro numero: ');
    readln(n);
    if (n>may) then begin
      may:=n;
    end;
  end;
  writeln('El numero mayor es: ', may);
end.
  
```



Ejercicio N° 27

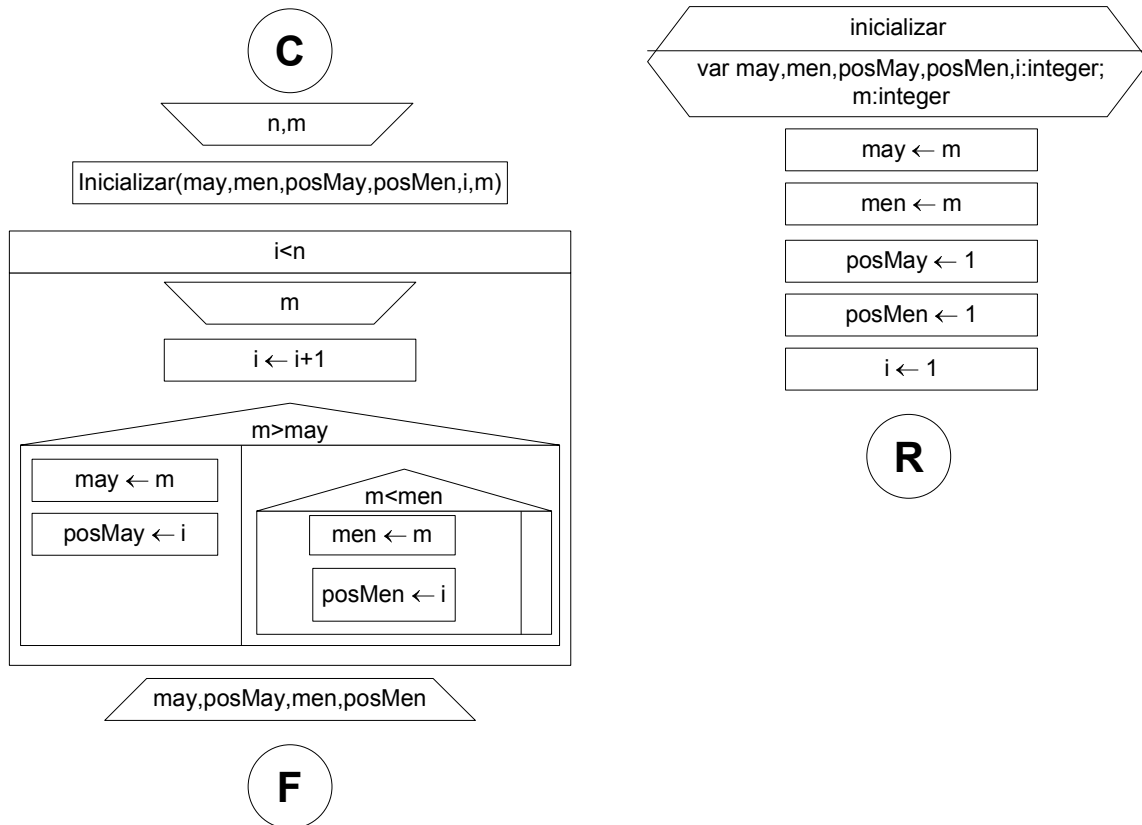
Dados N valores informar el mayor, el menor y en que posición del conjunto fueron ingresados.

```

procedure inicializar(var may,men,posMay,posMen,i:integer;
m:integer);
begin
    may:=m;
    men:=m;
    posMay:=1;
    posMen:=1;
    i:=1
end;

var n,m,may,men,posmay,posmen,i:integer;
begin
    writeln('Ingrese la cantidad de numeros que va a ingresar
para calcular:');
    read(n);
    writeln('Ingrese un numero:');
    read(m);
  
```

```
inicializar(may,men,posMay,posMen,i,m);
while (i<n) do begin
    writeln('Ingrese un numero:');
    read(m);
    i:=i+1;
    if (m>may) then begin
        may:=m;
        posMay:=i;
    end else begin
        if (m<men) then begin
            men:=m;
            posMen:=i;
        end;
    end;
end;
writeln('Numero mayor:',may);
writeln('Posicion del numero mayor:',posMay);
writeln('Numero menor:',men);
writeln('Posicion del numero menor:',posMen);
end.
```



Ejercicio N° 28

Dado un conjunto de Nombres y Fechas de nacimientos (AAAAMMDD), que finaliza con un Nombre = 'FIN', informar el nombre de la persona con mayor edad y el de la más joven.

```
function getAnio(fec:longint):integer;
begin
    getAnio:=fec div 10000;
end;
```

```
function getMes(fec:longint):integer;
var x:integer;
begin
    x:=fec mod 10000;
    getMes:= x div 100;
end;
```

```
function getDia(fec:longint):integer;
begin
    getDia:= fec mod 100;
```

```
end;
```

```
procedure fechas(fec,fecMay,fecMen:longint; var  
anio,anioMen,anioMay,mes,mesMen,mesMay,dia,diaMen,diaMay:integer  
);
```

```
begin
```

```
    anio:=getAnio(fec);
```

```
    anioMen:=getAnio(fecMen);
```

```
    anioMay:=getAnio(fecMay);
```

```
    mes:=getMes(fec);
```

```
    mesMen:=getMes(fecMen);
```

```
    mesMay:=getMes(fecMay);
```

```
    dia:=getDia(fec);
```

```
    diaMen:=getDia(fecMen);
```

```
    diaMay:=getDia(fecMay);
```

```
end;
```

```
procedure aniosDistintos(var perMay,perMen:string[15];var  
fecMay,fecMen:longint; fec:longint;nom:string[15]);
```

```
var
```

```
anio,anioMen,anioMay,mes,mesMen,mesMay,dia,diaMen,diaMay:integer  
;
```

```
begin
```

```
    fechas(fec,fecMay,fecMen,anio,anioMen,anioMay,mes,mesMen,mesMay,dia,diaMen,diaMay);
```

```
    if (anio>anioMen) then begin
```

```
        fecMen:=fec;
```

```
        perMen:=nom;
```

```
    end else begin
```

```
        if (anio<anioMay) then begin
```

```
            fecMay:=fec;
```

```
            perMay:=nom;
```

```
        end;
```

```
    end;
```

```
end;
```

```
procedure mesesDistintos(var perMay,perMen:string[15];var  
fecMay,fecMen:longint; fec:longint;nom:string[15]);
```

```

var
anio,anioMen,anioMay,mes,mesMen,mesMay,dia,diaMen,diaMay:integer
;
begin
    fechas(fec,fecMay,fecMen,anio,anioMen,anioMay,mes,mesMen,m
esMay,dia,diaMen,diaMay);
    if ((anio=anioMen) or (anio=anioMay)) then begin
        if (mes>mesMen) then begin
            fecMen:=fec;
            perMen:=nom;
        end else begin
            if (mes<mesMay) then begin
                fecMay:=fec;
                perMay:=nom;
            end;
        end;
    end;
end;

procedure diasDistintos(var perMay,perMen:string[15];var
fecMay,fecMen:longint; fec:longint;nom:string[15]);
var
anio,anioMen,anioMay,mes,mesMen,mesMay,dia,diaMen,diaMay:integer
;
begin
    fechas(fec,fecMay,fecMen,anio,anioMen,anioMay,mes,mesMen,m
esMay,dia,diaMen,diaMay);
    if (((anio=anioMen) and (mes=mesMen)) or ((anio=anioMay)
and (mes=mesMay))) then begin
        if (dia>diaMen) then begin
            fecMen:=fec;
            perMen:=nom;
        end else begin
            if (dia<diaMay) then begin
                fecMay:=fec;
                perMay:=nom;
            end;
        end;
    end;
end;

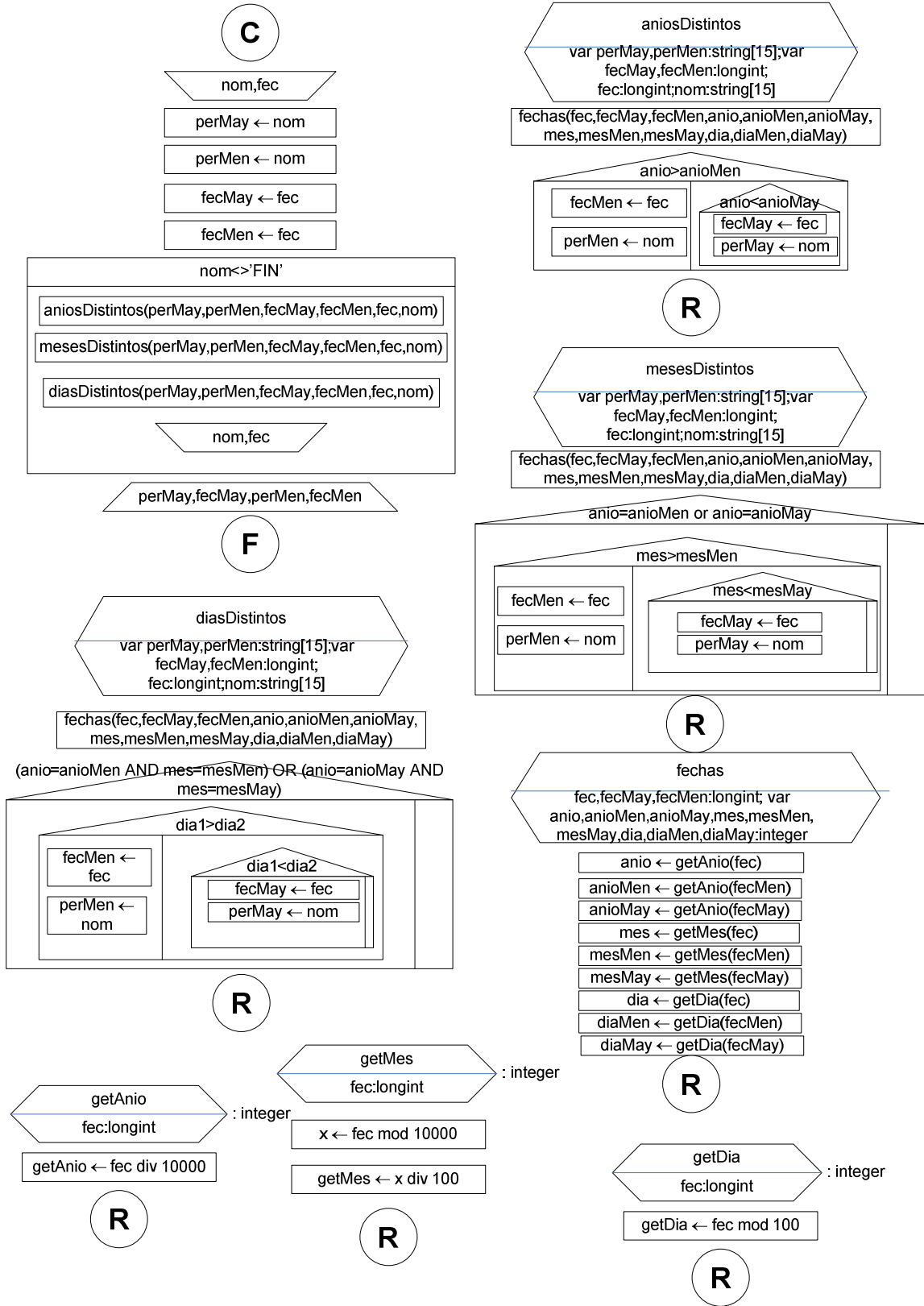
```

```

end;

//Programa principal
var nom,perMen,perMay:string[15]; fec,fecMen,fecMay:longint;
begin
    writeln('Ingrese un nombre:');
    read(nom);
    writeln('Ingrese la fecha de nacimiento (AAAAMMDD):');
    read(fec);
    perMen:=nom;
    perMay:=nom;
    fecMen:=fec;
    fecMay:=fec;
    while (nom<>'FIN') do begin
        aniosDistintos(perMay,perMen,fecMay,fecMen,fec,nom);
        mesesDistintos(perMay,perMen,fecMay,fecMen,fec,nom);
        diasDistintos(perMay,perMen,fecMay,fecMen,fec,nom);
        writeln('Ingrese un nombre:');
        readln(nom);
        writeln('Ingrese la fecha de nacimiento
(AAAAMMDD):');
        readln(fec);
    end;
    writeln(perMen, ' es la persona de menor edad, nacido el
',fecMen);
    writeln(perMay, ' es la persona de mayor edad, nacido el
',fecMay);
end.

```



Ejercicio N° 29

Se dispone de un lote de valores enteros positivos que finaliza con un número negativo. El lote está dividido en sublotes por medio de valores cero. Desarrollar un programa que determine e informe:

1. por cada sublote el promedio de valores
2. el total de sublotes procesados
3. el valor máximo del conjunto, indicando en que sublote se encontró y la posición relativa del mismo dentro del sublote
4. valor mínimo de cada sublote

Nota: el lote puede estar vacío (primer valor negativo), o puede haber uno, varios o todos los sublotes vacíos (ceros consecutivos).

```
procedure inicializar(var max,contSub,maxSub,maxPos:integer;
n:integer);
begin
    max:=n;
    contSub:=0;
    maxSub:=1;
    maxPos:=1;
end;
```

```
procedure cabLote(var sum,cont,min:integer; n:integer);
begin
    sum:=0;
    cont:=0;
    min:=n;
end;
```

```
procedure proceso(var
sum,cont,max,maxSub,maxPos,min,contSub:integer; n:integer);
begin
    sum:=sum+n;
    cont:=cont+1;
    if ((n>max) and (n>0))then begin
        max:=n;
        maxSub:=contSub+1;
```

```

        maxPos:=cont;
    end else begin
        if ((n<min) and (n>0))then begin
            min:=n;
        end;
    end;
end;

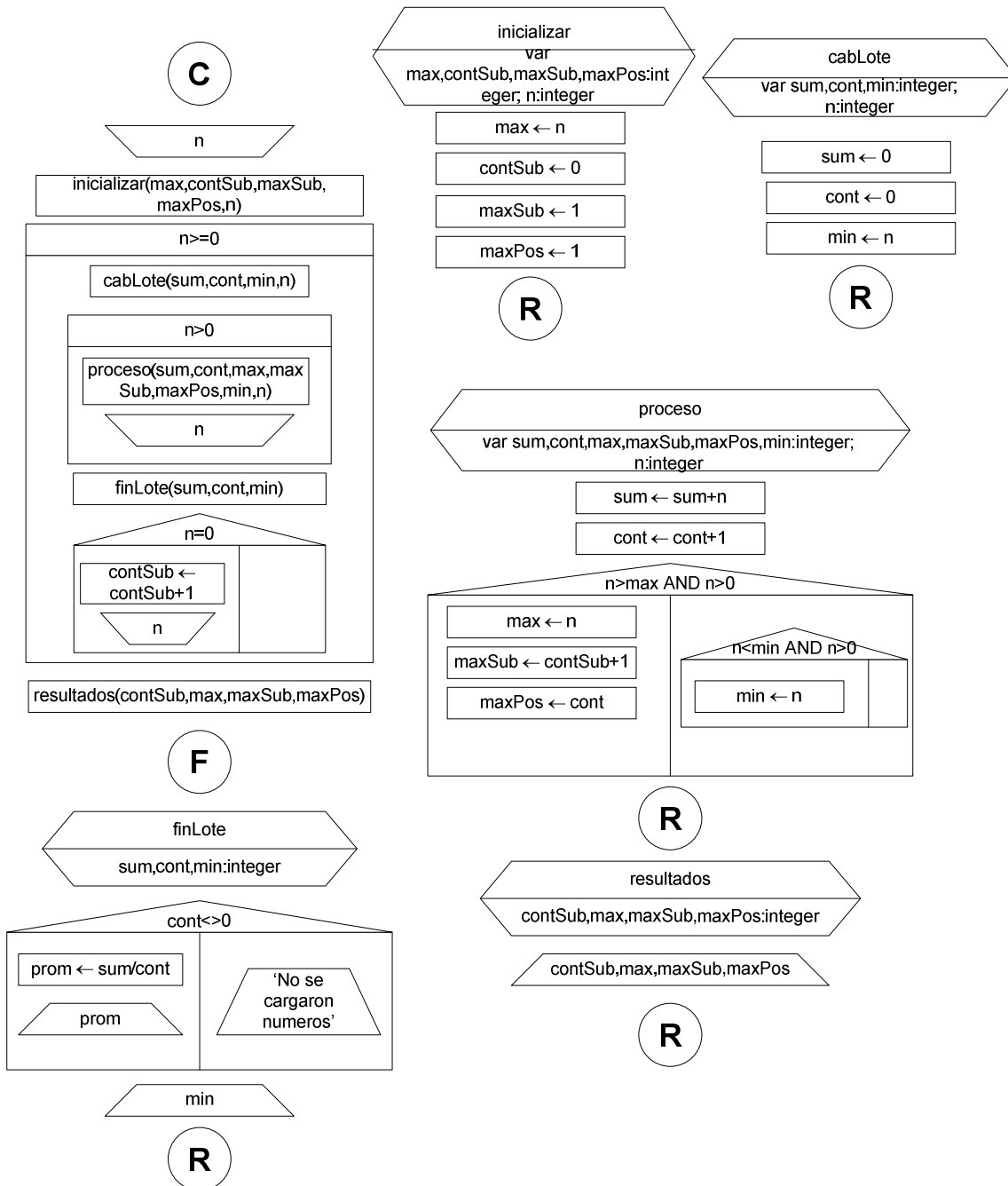
procedure finLote(sum,cont,min:integer);
var prom:real;
begin
    if (cont>0) then begin
        prom:=(sum/cont);
        writeln('El promedio del sublote es: ',prom);
    end else begin
        writeln('No se cargaron datos');
    end;
    writeln('El valor minimo del sublote es: ',min);
end;

procedure resultados(contSub,max,maxSub,maxPos:integer);
begin
    writeln('Cantidad de sublotes: ',contSub);
    writeln('El valor maximo del conjunto: ',max);
    writeln('Sublote donde se encuentra el valor maximo: ',maxSub);
    writeln('Posicion en el sublote donde se encuentra el valor maximo: ',maxPos);
end;

//programa principal
var n,max,contSub,maxSub,maxPos,sum,cont,min:integer;
begin
    writeln('Ingrese un numero: ');
    read(n);
    inicializar(max,contSub,maxSub,maxPos,n);
    while (n>=0) do begin
        cabLote(sum,cont,min,n);
        while (n>0) do begin

```

```
    proceso(sum, cont, max, maxSub, maxPos, min, contSub, n);
        writeln('Ingrese un numero: ');
        read(n);
    end;
    finLote(sum, cont, min);
    if (n=0) then begin
        contSub:=contSub+1;
        writeln('Ingrese un numero: ');
        read(n);
    end;
end;
resultados(contSub, max, maxSub, maxPos);
end.
```



Ejercicio N° 30

Dada una serie de M pares {color, número} que corresponden a los tiros de una ruleta. Se pide informar:

1. cuantas veces salió el número cero y el número anterior a cada cero
2. cuantas veces seguidas llegó a repetirse el color negro
3. cuantas veces seguidas llegó a repetirse el mismo número y cual fue

4. el mayor número de veces seguidas que salieron alternados el rojo y el negro
5. el mayor número de veces seguidas que se negó la segunda docenas

```
procedure inicializar(var
contCero,negRep,vecSeg,maxVecSeg,nAnt,maxVecSegN,vecSegRN,maxVec
SegRN,vecSegDoc,maxVecSegDoc:integer);
begin
    contCero:=0;
    negRep:=1;
    vecSeg:=1;
    maxVecSeg:=0;
    maxVecSegN:=-1;
    nAnt:=-1;
    vecSegRN:=0;
    maxVecSegRN:=-1;
    vecSegDoc:=0;
    maxVecSegDoc:=-1;
end;

procedure nroCero(n,nAnt,i:integer; var contCero:integer);
begin
    if (n=0) then begin
        contCero:=contCero+1;
        if (i>0) then begin
            writeln('Numero anterior al cero: ',nAnt);
        end;
    end;
end;

procedure colNegro(col,cAnt:string[10];i:integer;var
negRep,maxVecSegN:integer);
begin
    if ((col='negro') and (i>1)) then begin
        if (cAnt=col) then begin
            negRep:=negRep+1;
        end else begin
            negRep:=1;
        end;
    end;
end;
```

```

        if (negRep>maxVecSegN) then begin
            maxVecSegN:=negRep;
        end;
    end;

procedure nroRepet(n,nAnt:integer; var
vecSeg,maxVecSeg,maxNroRep:integer);
begin
    if (n=nAnt) then begin
        vecSeg:=vecSeg+1;
    end else begin
        vecSeg:=1;
    end;
    if (vecSeg>maxVecSeg) then begin
        maxVecSeg:=vecSeg;
        maxNroRep:=n;
    end;
end;

procedure maxRojoNegro(col,cAnt:string[10]; var
vecSegRN,maxVecSegRN:integer);
begin
    if (col='rojo') then begin
        if (cAnt='negro') then begin
            vecSegRN:=vecSegRN+1;
        end;
    end else if (col='negro') then begin
        if (cAnt='rojo') then begin
            vecSegRN:=vecSegRN+1;
        end;
    end else begin
        vecSegRN:=0;
    end;
    if (vecSegRN>maxVecSegRN) then begin
        maxVecSegRN:=vecSegRN;
    end;
end;

```

```

procedure maxSegDoc(n:integer; var
vecSegDoc,maxVecSegDoc:integer);
begin
    if ((n>0) and (n<=24)) then begin
        vecSegDoc:=vecSegDoc+1;
    end else begin
        vecSegDoc:=0;
    end;
    if (vecSegDoc>maxVecSegDoc) then begin
        maxVecSegDoc:=vecSegDoc;
    end;
end;

procedure
resultados(contCero,maxVecSeg,maxNroRep,maxVecSegRN,maxVecSegDoc
,maxVecSegN:integer);
begin
    writeln('El numero cero salio ',contCero,' veces');
    writeln('La mayor cantidad de veces seguidas que salio el
color negro: ',maxVecSegN);
    writeln('El numero ',maxNroRep,' es el que salio mas veces
seguidas');
    writeln('El mayor numero de veces seguidas que salieron
alternadas el rojo y el negro: ', maxVecSegRN);
    writeln('El mayor numero de veces seguidas que se nego la
segunda docena: ',maxVecSegDoc);
end;

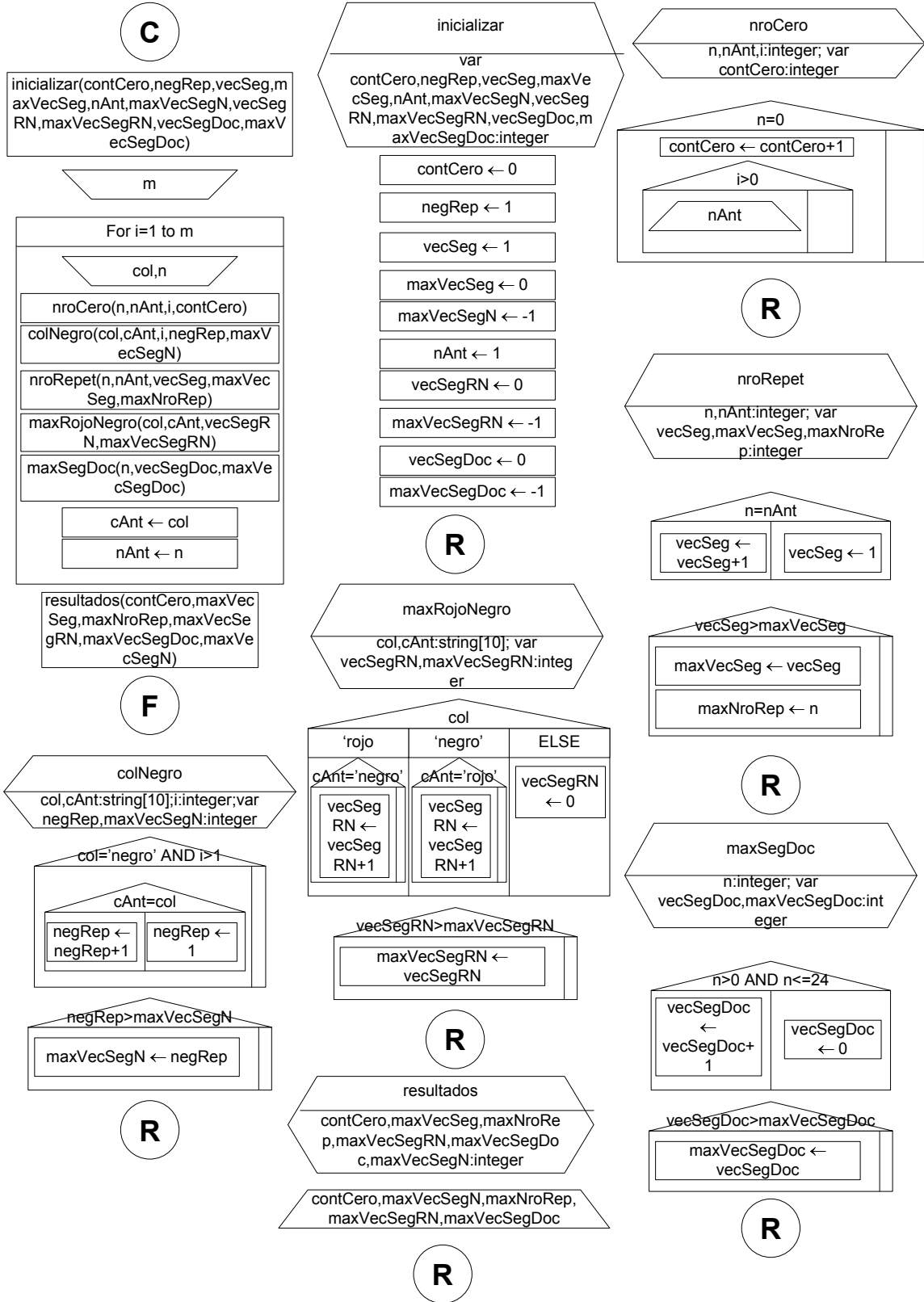
//programa principal
var
m,contCero,negRep,vecSeg,maxVecSeg,n,nAnt,i,maxVecSegN,maxNroRep
,vecSegRN,maxVecSegRN,vecSegDoc,maxVecSegDoc:integer;
col,cAnt:string[10];
begin
    inicializar(contCero,negRep,vecSeg,maxVecSeg,nAnt,maxVecSe
gN,vecSegRN,maxVecSegRN,vecSegDoc,maxVecSegDoc);
    writeln('Ingrese la cantidad de pares que quiere cargar:
');
    read(m);

```

```

    for i:=1 to m do begin
        writeln('Ingrese el color: ');
        read(col);
        writeln('Ingrese el numero: ');
        read(n);
        nroCero(n,nAnt,i,contCero);
        colNegro(col,cAnt,i,negRep,maxVecSegN);
        nroRepet(n,nAnt,vecSeg,maxVecSeg,maxNroRep);
        maxRojoNegro(col,cAnt,vecSegRN,maxVecSegRN);
        maxSegDoc(n,vecSegDoc,maxVecSegDoc);
        cAnt:=col;
        nAnt:=n;
    end;
    resultados(contCero,maxVecSeg,maxNroRep,maxVecSegRN,maxVec
SegDoc,maxVecSegN);
end.

```



Ejercicio N° 31

Dada una serie de caracteres que conforman una oración, donde cada palabra está separada de la siguiente por un carácter blanco y la oración finaliza con un punto. Se pide informar:

1. cantidad de veces que apareció cada vocal
2. cantidad de palabras que contiene la oración
3. cantidad de letras que posee la palabra más larga

```
procedure inicializar(var pal,maxPal,contPal,i:integer; var
contA,contE,contI,contO,contU:integer);
begin
    pal:=0;
    maxPal:=0;
    contPal:=1;
    contA:=0;
    contE:=0;
    contI:=0;
    contO:=0;
    contU:=0;
end;
```

```
procedure contVocal(carac:string[1]; var
contA,contE,contI,contO,contU:integer);
begin
    if carac='a' then begin
        contA:=contA+1
    end;
    if carac='e' then begin
        contE:=contE+1
    end;
    if carac='i' then begin
        contI:=contI+1
    end;
    if carac='o' then begin
        contO:=contO+1
    end;
    if carac='u' then begin
```

```

        contU:=contU+1
    end;
end;

procedure contPalabra(carac:string[1];var contPal:integer);
begin
    if (carac=' ') then begin
        contPal:=contPal+1;
    end;
end;

procedure maxPalabra(carac:string[1];var pal,maxPal:integer);
begin
    if ((carac<>' ') and (carac<>'.')) then begin
        pal:=pal+1;
    end else begin
        pal:=0;
    end;
    if (pal>maxPal) then begin
        maxPal:=pal;
    end;
end;

procedure resultados(contA,contE,contI,contO,contU:integer;
contPal,maxPal:integer);
begin
    writeln('Veces que aparecio la letra A: ',contA);
    writeln('Veces que aparecio la letra E: ',contE);
    writeln('Veces que aparecio la letra I: ',contI);
    writeln('Veces que aparecio la letra O: ',contO);
    writeln('Veces que aparecio la letra U: ',contU);
    writeln('Cantidad de palabras que contiene la oracion:
',contPal);
    writeln('Cantidad de letras que posee la palabra mas
larga: ',maxPal);
end;

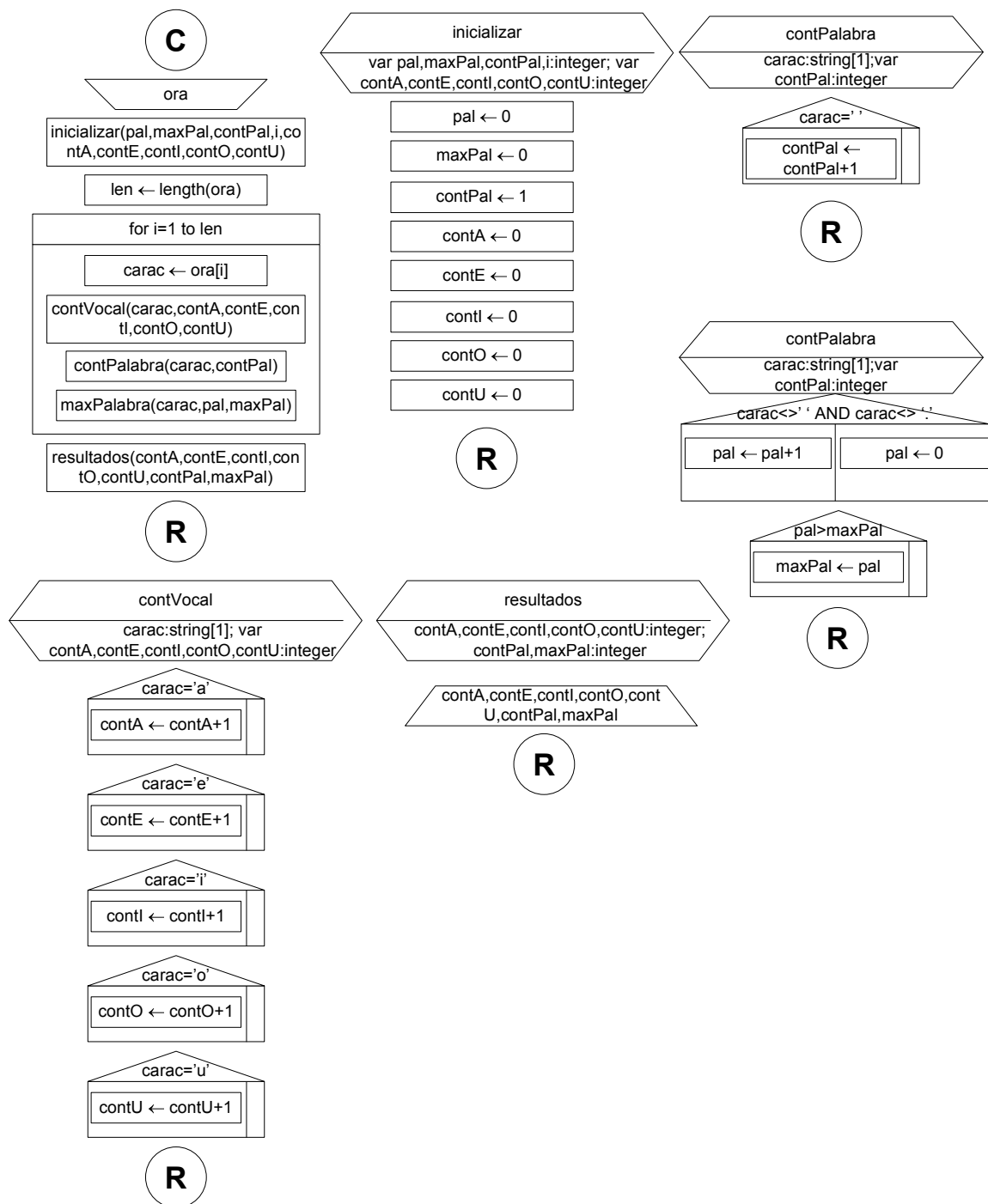
//programa principal

```

```

var ora:string[50];
pal,maxPal,contPal,i,len,contA,contE,contI,contO,contU:integer;
carac:string[1];
begin
    writeln('Escriba una oracion: ');
    read(ora);
    inicializar(pal,maxPal,contPal,i,contA,contE,contI,contO,c
ontU);
    len:=length(ora);
    for i:=1 to len do begin
        carac:=ora[i];
        contVocal(carac,contA,contE,contI,contO,contU);
        contPalabra(carac,contPal);
        maxPalabra(carac,pal,maxPal);
    end;
    resultados(contA,contE,contI,contO,contU,contPal,maxPal);
end.

```



Ejercicio N° 32

Dado un número entero positivo entre 1 y 3999 informar su correspondiente número Romano.

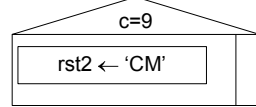
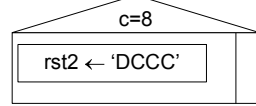
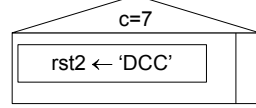
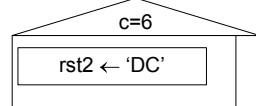
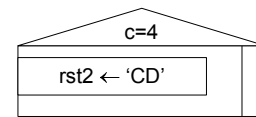
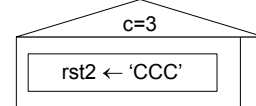
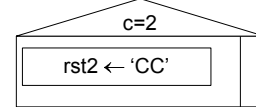
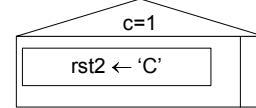
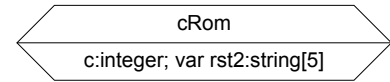
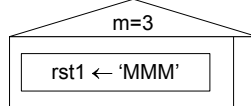
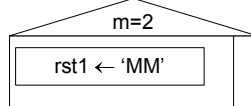
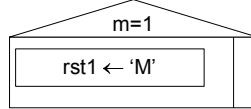
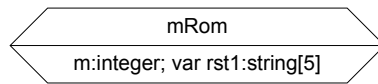
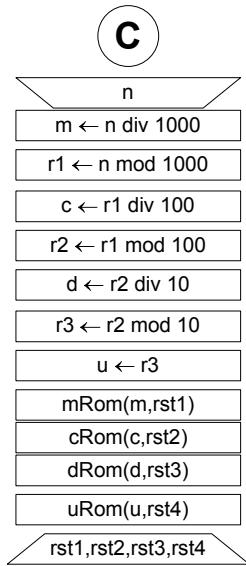
```
var n: integer;
```

```

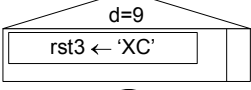
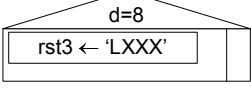
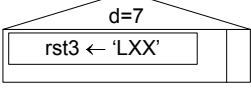
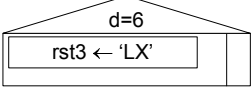
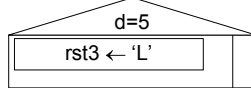
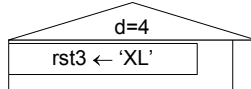
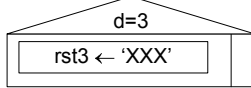
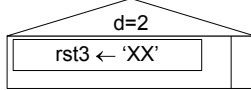
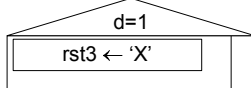
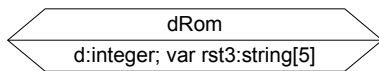
var m, c, d, u, r1, r2, r3: integer;
var rst1, rst2, rst3, rst4: string[5];
begin
  writeln('Ingrese un valor: ');
  readln(n);
  m:= n div 1000;
  r1:= n mod 1000;
  c:= r1 div 100;
  r2:= r1 mod 100;
  d:= r2 div 10;
  r3:= r2 mod 10;
  u:= r3;
  case m of
    1: rst1:= 'M';
    2: rst1:= 'MM';
    3: rst1:= 'MMM';
  end;
  case c of
    1: rst2:= 'C';
    2: rst2:= 'CC';
    3: rst2:= 'CCC';
    4: rst2:= 'CD';
    5: rst2:= 'D';
    6: rst2:= 'DC';
    7: rst2:= 'DCC';
    8: rst2:= 'DCCC';
    9: rst2:= 'CM';
  end;
  case d of
    1: rst3:= 'X';
    2: rst3:= 'XX';
    3: rst3:= 'XXX';
    4: rst3:= 'XL';
    5: rst3:= 'L';
    6: rst3:= 'LX';
    7: rst3:= 'LXX';
    8: rst3:= 'LXXX';
    9: rst3:= 'XC';
  end;
end;

```

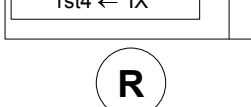
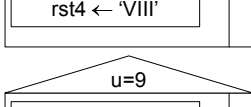
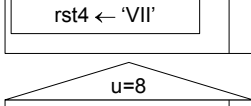
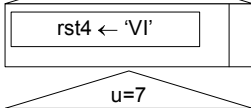
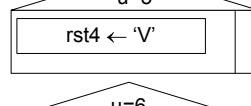
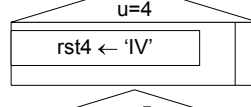
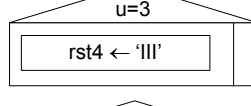
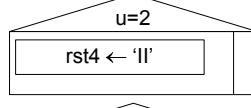
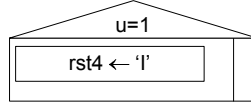
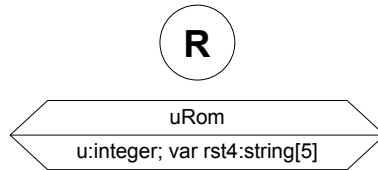
```
case u of
  1: rst4:= 'I';
  2: rst4:= 'II';
  3: rst4:= 'III';
  4: rst4:= 'IV';
  5: rst4:= 'V';
  6: rst4:= 'VI';
  7: rst4:= 'VII';
  8: rst4:= 'VIII';
  9: rst4:= 'IX';
end;
writeln('El numero romano es:',rst1, rst2, rst3, rst4);
end.
```



F



R



R

R

Ejercicio N° 33

Dado un conjunto de valores enteros, calcular e informar a) cuantos valores cero hubo, b) promedio de valores positivos, c) sumatoria de valores negativos.

Resolver el ejercicio para los siguientes lotes de datos:

1. 167 valores enteros
2. N valores, donde el valor de N debe ser leído previamente
3. El conjunto de valores termina con un valor igual anterior
4. Se dan N valores, pero el proceso deberá finalizar si se procesan todos los valores o si la cantidad de ceros supera a cuatro
5. Se dan N valores, pero el proceso deberá finalizar si se cumple alguna de las condiciones del punto 4 o si el promedio de positivos resulta mayor que seis

1-

```
procedure inicializar(var
contCero,sumPos,contPos,sumNeg:integer);
begin
    contCero:=0;
    sumPos:=0;
    contPos:=0;
    sumNeg:=0;
end;

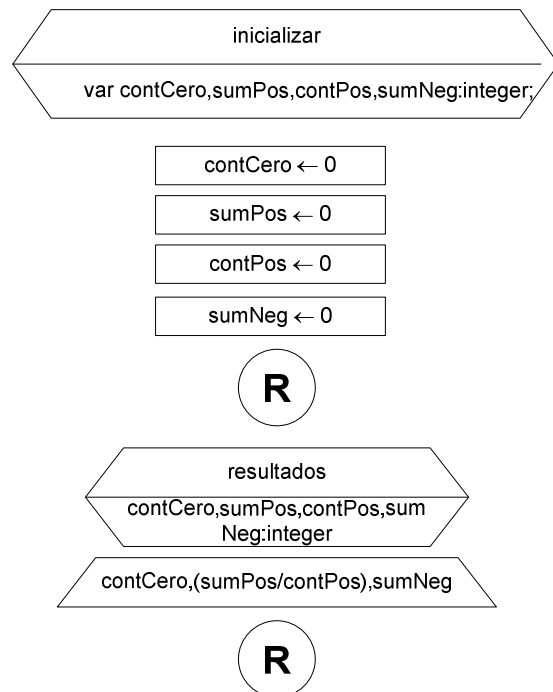
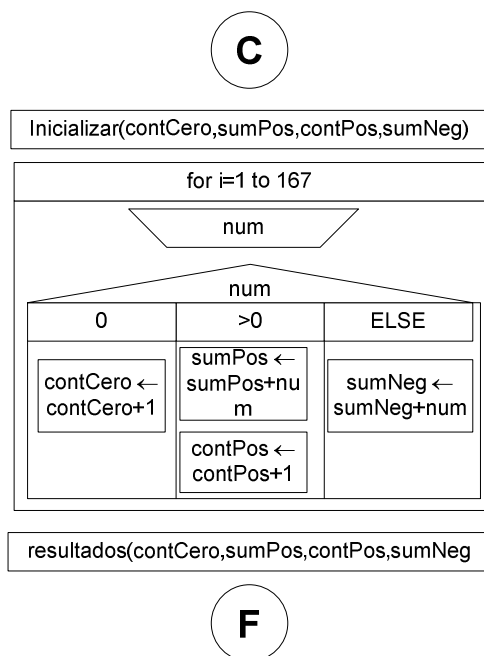
procedure resultados(contCero,sumPos,contPos,sumNeg:integer);
begin
    writeln('Cantidad de veces que fue cargado el cero:
',contCero);
    writeln('Promedio de valores positivos:
',(sumPos/contPos));
    writeln('Sumatoria de valores negativos: ',sumNeg);
end;

var num,contCero,sumPos,contPos,sumNeg,i:integer;
begin
    inicializar(contCero,sumPos,contPos,sumNeg);
    for i:=1 to 167 do begin
        writeln('Ingrese un numero: ');
        read(num);
```

```

    if (num=0) then begin
        contCero:=contCero+1;
    end else if (num>0) then begin
        sumPos:=sumpos+num;
        contPos:=contPos+1;
    end else begin
        sumNeg:=sumNeg+num;
    end;
end;
resultados(contCero, sumPos, contPos, sumNeg);
end.

```



2-

```

procedure inicializar(var
contCero, sumPos, contPos, sumNeg:integer);
begin
    contCero:=0;
    sumPos:=0;
    contPos:=0;
    sumNeg:=0;
end;

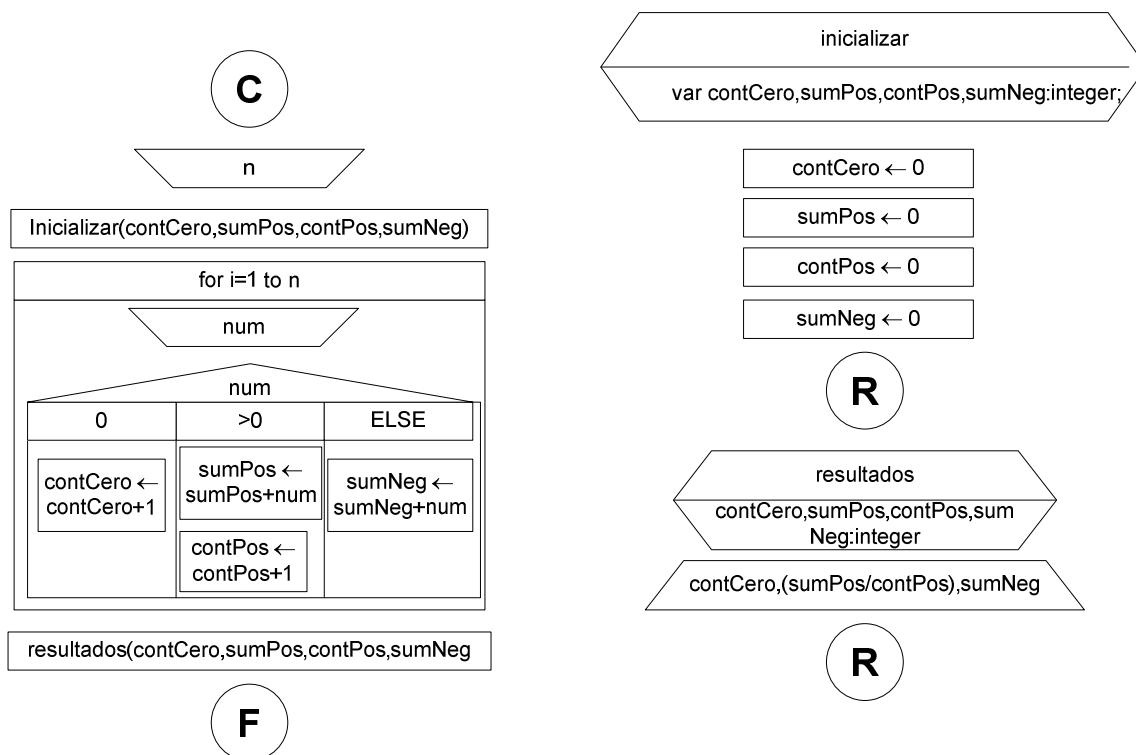
```

```

procedure resultados(contCero,sumPos,contPos,sumNeg:integer);
begin
    writeln('Cantidad de veces que fue cargado el cero:
',contCero);
    writeln('Promedio de valores positivos:
',(sumPos/contPos));
    writeln('Sumatoria de valores negativos: ',sumNeg);
end;

var n,num,contCero,sumPos,contPos,sumNeg,i:integer;
begin
    inicializar(contCero,sumPos,contPos,sumNeg);
    writeln('Ingrese la cantidad de numeros que va a ingresar:
');
    read(n);
    for i:=1 to n do begin
        writeln('Ingrese un numero: ');
        read(num);
        if (num=0) then begin
            contCero:=contCero+1;
        end else if (num>0) then begin
            sumPos:=sumpos+num;
            contPos:=contPos+1;
        end else begin
            sumNeg:=sumNeg+num;
        end;
    end;
    resultados(contCero,sumPos,contPos,sumNeg);
end.

```



3-

```

procedure inicializar(var
contCero, sumPos, contPos, sumNeg:integer);
begin

```

```

    contCero:=0;

```

```

    sumPos:=0;

```

```

    contPos:=0;

```

```

    sumNeg:=0;

```

```

end;

```

```

procedure resultados(contCero,sumPos,contPos,sumNeg:integer);
begin

```

```

    writeln('Cantidad de veces que fue cargado el cero:
',contCero);

```

```

    writeln('Promedio de valores positivos:
',(sumPos/contPos));

```

```

    writeln('Sumatoria de valores negativos: ',sumNeg);
end;

```

```

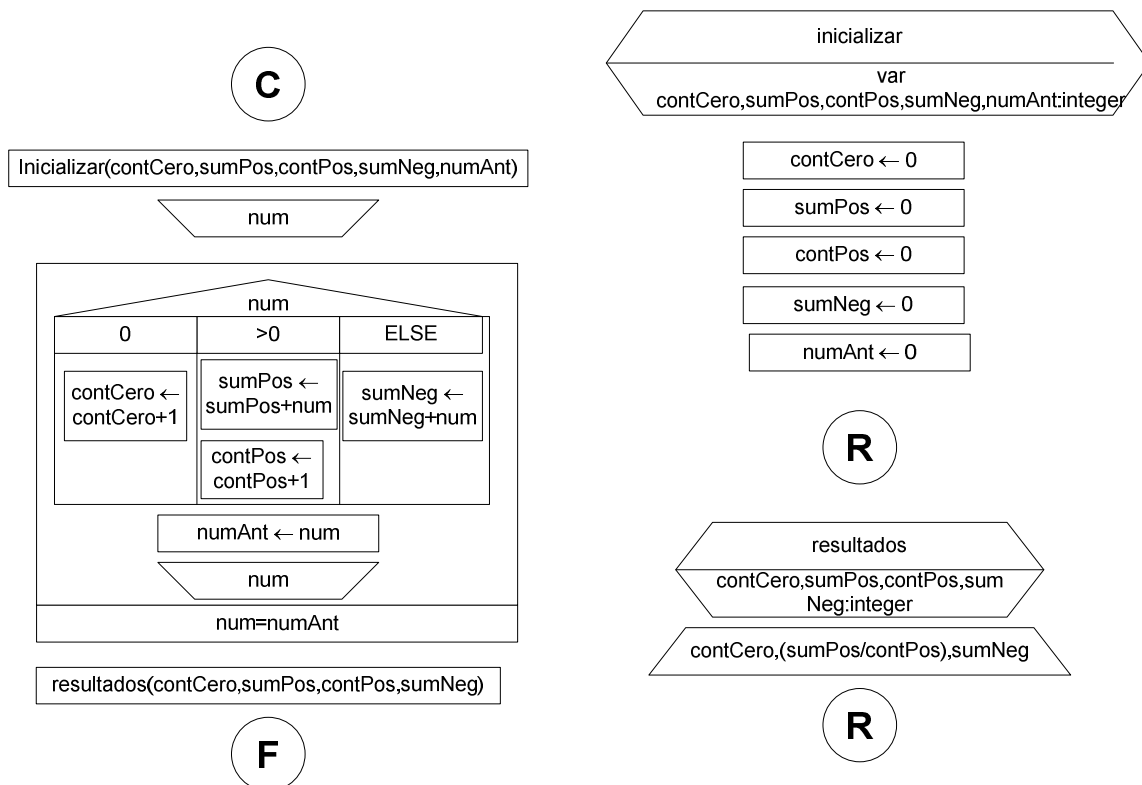
var num, contCero, sumPos, contPos, sumNeg, i:integer;

```

```

begin
  inicializar(contCero,sumPos,contPos,sumNeg);
  writeln('Ingrese un numero: ');
  read(num);
  repeat
    if (num=0) then begin
      contCero:=contCero+1;
    end else if (num>0) then begin
      sumPos:=sumpos+num;
      contPos:=contPos+1;
    end else begin
      sumNeg:=sumNeg+num;
    end;
    numAnt:=num;
    writeln('Ingrese un numero: ');
    read(num);
  until (num=numAnt);
  resultados(contCero,sumPos,contPos,sumNeg);
end.

```



4-

```
procedure inicializar(var
contCero,sumPos,contPos,sumNeg,i:integer);
begin
    contCero:=0;
    sumPos:=0;
    contPos:=0;
    sumNeg:=0;
    i:=0;
end;

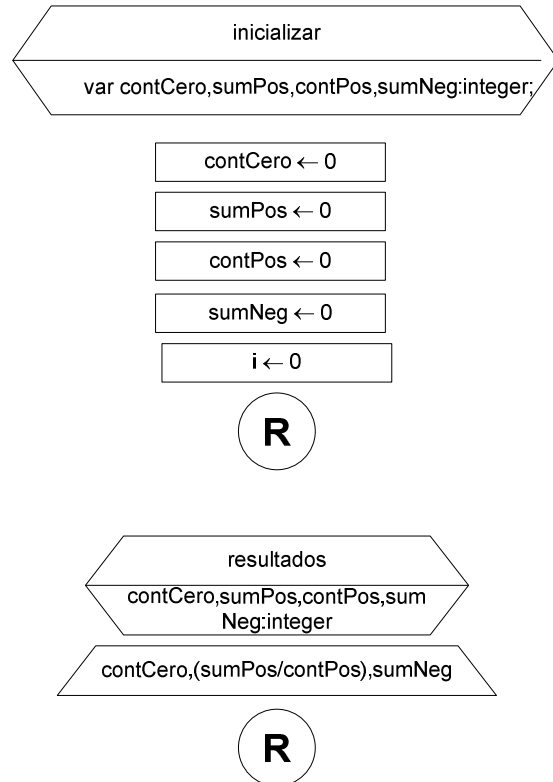
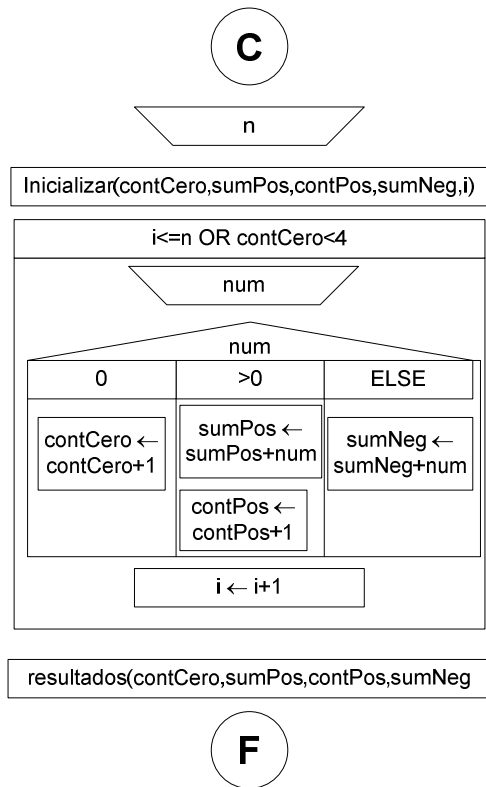
procedure resultados(contCero,sumPos,contPos,sumNeg:integer);
begin
    writeln('Cantidad de veces que fue cargado el cero:
',contCero);
    writeln('Promedio de valores positivos:
',(sumPos/contPos));
    writeln('Sumatoria de valores negativos: ',sumNeg);
end;

var n,num,contCero,sumPos,contPos,sumNeg,i:integer;
begin
    inicializar(contCero,sumPos,contPos,sumNeg,i);
    writeln('Ingrese la cantidad de numeros que va a ingresar:
');
    read(n);
    while ((i<=n) or (contCero<4)) do begin
        writeln('Ingrese un numero: ');
        read(num);
        if (num=0) then begin
            contCero:=contCero+1;
        end else if (num>0) then begin
            sumPos:=sumpos+num;
            contPos:=contPos+1;
        end else begin
            sumNeg:=sumNeg+num;
        end;
        i:=i+1;
    end;
end;
```

```

    resultados(contCero, sumPos, contPos, sumNeg);
end.

```



5-

```

procedure inicializar(var
contCero, sumPos, contPos, sumNeg, i:integer);
begin
    contCero:=0;
    sumPos:=0;
    contPos:=0;
    sumNeg:=0;
    i:=0;
end;

```

```

procedure resultados(contCero, sumPos, contPos, sumNeg:integer);
begin
    writeln('Cantidad de veces que fue cargado el cero:
',contCero);

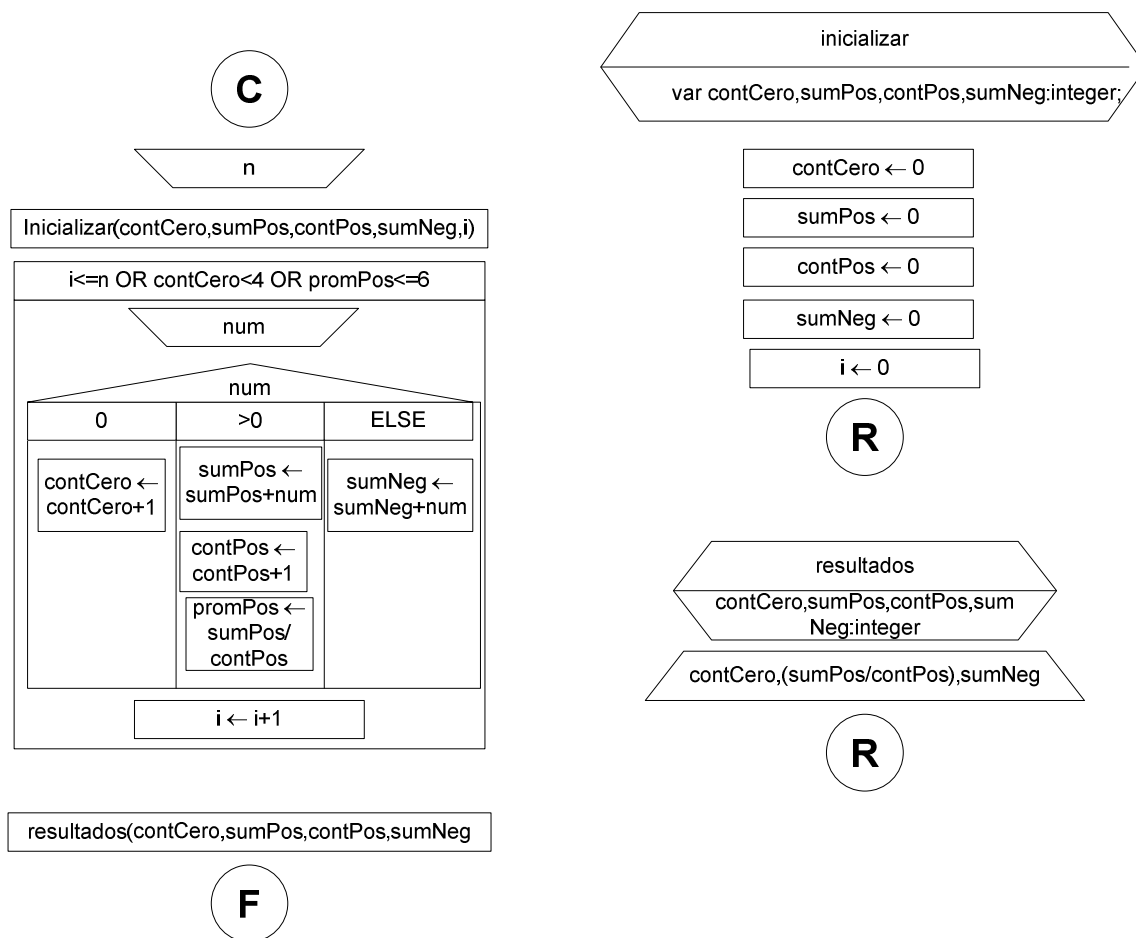
```

```

        writeln('Promedio de valores positivos:
', (sumPos/contPos));
        writeln('Sumatoria de valores negativos: ', sumNeg);
end;

var n, num, contCero, sumPos, contPos, sumNeg, i: integer; promPos: real;
begin
    inicializar(contCero, sumPos, contPos, sumNeg, i);
    writeln('Ingrese la cantidad de numeros que va a ingresar:
');
    read(n);
    while ((i<=n) or (contCero<4) or (promPos<=6)) do begin
        writeln('Ingrese un numero: ');
        read(num);
        if (num=0) then begin
            contCero:=contCero+1;
        end else if (num>0) then begin
            sumPos:=sumPos+num;
            contPos:=contPos+1;
            promPos:=(sumPos/contPos);
        end else begin
            sumNeg:=sumNeg+num;
        end;
        i:=i+1;
    end;
    resultados(contCero, sumPos, contPos, sumNeg);
end.

```



Ejercicio N° 34

El gobierno de la Ciudad de Buenos Aires realiza una encuesta en casas de familia. De cada familia conoce: domicilio, tipo de vivienda ('C':casa, 'D':departamento), y cantidad de integrantes.

De cada integrante de la familia se conoce: nombre y apellido, edad, sexo ('F', 'M'), nivel de estudios alcanzados ('N': no posee, 'P': primario, 'S': secundario, 'T': terciario, 'U': universitario), y un indicador ('I': incompleto, 'C': completo) que se refiere al ítem anterior.

Los datos finalizan cuando la cantidad de integrantes sea igual a cero.

Se pide emitir un listado con los resultados:

1. los datos de los encuestados que hayan completado los estudios primarios
2. el porcentaje de analfabetismo en la ciudad (se considera analfabetos a los mayores de 10 años que no posean estudios)
3. el domicilio de la familia con mayor cantidad de integrantes que viven en departamento

4. edad promedio de cada familia y de la ciudad
5. cantidad de encuestados en cada tipo de nivel de estudios alcanzados incompletos.
6. Porcentaje de encuestados de sexo femenino y masculino

```
procedure inicializar(var  
contIncP,contIncS,contIncT,contIncU,contFem,contMas,sumEdCiu,con  
tCiu,maxInt,contAnal:integer);
```

```
begin
```

```
    contIncP:=0;
```

```
    contIncS:=0;
```

```
    contIncT:=0;
```

```
    contIncU:=0;
```

```
    contFem:=0;
```

```
    contMas:=0;
```

```
    sumEdCiu:=0;
```

```
    contCiu:=0;
```

```
    maxInt:=0;
```

```
    contAnal:=0;
```

```
end;
```

```
procedure maxIntDom(dom:string[20];int:integer;var  
maxDom:string[20]; var maxInt:integer);
```

```
begin
```

```
    if (int>maxInt) then begin
```

```

        maxInt:=int;

        maxDom:=dom;

    end;

end;

procedure cabInt(var sumEdFlia,contEdFlia,int:integer; var
intAnt,maxInt:integer; var maxDom:string[20]; dom:string[20]);

begin

    sumEdFlia:=0;

    contEdFlia:=0;

    maxIntDom(dom,int,maxDom,maxInt);

    intAnt:=int;

end;

procedure
estPrim(est,ind,sexo:string[1];nom,ape:string[20];edad:integer);

begin

    if ((est='P') and (ind='c')) then begin

        writeln('Nombre: ',nom);

        writeln('Apellido: ',ape);

        writeln('Edad: ',edad);

        writeln('Sexo: ',sexo);

    end;

end;

```

```

        end;

end;

procedure analfab(edad:integer;est:string[1];var
contAnal,contCiu:integer);

begin
    if ((edad>10) and (est='N')) then begin
        contAnal:=contAnal+1;

        end;

        contCiu:=contCiu+1
    end;

end;

procedure edadProm(edad:integer;var sumEdFlia:integer);

begin
    sumEdFlia:=sumEdFlia+edad;

end;

procedure contInc(ind,est:string[1]; var
contIncP,contIncS,contIncT,contIncU:integer);

begin
    if (ind='I') then begin
        if (est='P') then begin

```

```

        contIncP:=contIncP+1;
    end else if (est='S') then begin
        contIncS:=contIncS+1;
    end else if (est='T') then begin
        contIncT:=contIncT+1;
    end else if (est='U') then begin
        contIncU:=contIncU+1;
    end;
end;
end;

procedure contSexo(sexo:string[1]; var contFem,contMas:integer);
begin
    if (sexo='F') then begin
        contFem:=contFem+1;
    end else begin
        contMas:=contMas+1;
    end;
end;

procedure finInt(sumEdFlia,intAnt:integer;var sumEdCiu:integer);

```

```

begin

    sumEdCiu:=sumEdCiu+sumEdFlia;

    writeln('El promedio de las edades de la familia es:
',(sumEdFlia/intAnt));

end;


procedure
resultados(contAnal,contCiu,contFem,contMas,sumEdCiu,contIncP,co
ntIncS,contIncT,contIncU:integer;maxDom:string[20]);

var porcAna,porcFem,porcMas:real;

begin

    porcAna:=((contAnal*100)/contCiu);

    porcFem:=((contFem*100)/(contFem+contMas));

    porcMas:=((contMas*100)/(contFem+contMas));

    writeln('Porcentaje de analfabetismo en la ciudad:
',porcAna);

    writeln('Domicilio de la familia con mayor cantidad de
integrantes: ',maxDom);

    writeln('Edad promedio de la ciudad:
',(sumEdCiu/contCiu));

    writeln('Cantidad de encuestados con estudios primarios
incompletos: ', contIncP);

    writeln('Cantidad de encuestados con estudios secundarios
incompletos: ', contIncS);

```

```
        writeln('Cantidad de encuestados con estudios terciarios
incompletos: ', contIncT);
```

```
        writeln('Cantidad de encuestados con estudios
universitarios incompletos: ', contIncU);
```

```
        writeln('Porcentaje de encuestados de sexo femenino:
', porcFem);
```

```
        writeln('Porcentaje de encuestados de sexo masculino:
', porcMas);
```

```
end;
```

```
//programa principal
```

```
var
n,i,sumEdFlia,contEdFlia,int,intAnt,maxInt,edad,contAnal,contCiu
,contIncp,contIncS,contIncT,contIncU,contFem,contMas,sumEdCiu:in
teger;
```

```
var maxDom,dom,nom,ape:string[20]; est,ind,sexo,viv:string[1];
```

```
begin
```

```
        inicializar(contIncP,contIncS,contIncT,contIncU,contFem,co
ntMas,sumEdCiu,contCiu,maxInt,contAnal);
```

```
        writeln('Ingrese la cantidad de familias que desea
ingresar: ');
```

```
        read(n);
```

```
        for i:=1 to n do begin
```

```
                writeln('Ingrese el domicilio: ');
```

```
                read(dom);
```

```

        writeln('Ingrese el tipo de vivienda (C o D): ');
        read(viv);

        writeln('Ingrese la cantidad de integrantes: ');
        read(int);

        cabInt(sumEdFlia, contEdFlia, int, intAnt, maxInt, maxDom, dom);

        while (int>0) do begin
            writeln('Ingrese el nombre: ');
            read(nom);

            writeln('Ingrese el apellido: ');
            read(ape);

            writeln('Ingrese la edad: ');
            read(edad);

            writeln('Ingrese el sexo (F o M): ');
            read(sexo);

            writeln('Nivel de estudios alcanzados
(N,P,S,T,U): ');

            read(est);

            writeln('Indicador referido al item anterior
(I,C): ');

            read(ind);

            estPrim(est, ind, sexo, nom, ape, edad);

            analfab(edad, est, contAnal, contCiu);

```

```

        edadProm(edad, sumEdFlia);

    contInc(ind, est, contIncP, contIncS, contIncT, contIncU);

        contSexo(sexo, contFem, contMas);

        int:=int-1;

    end;

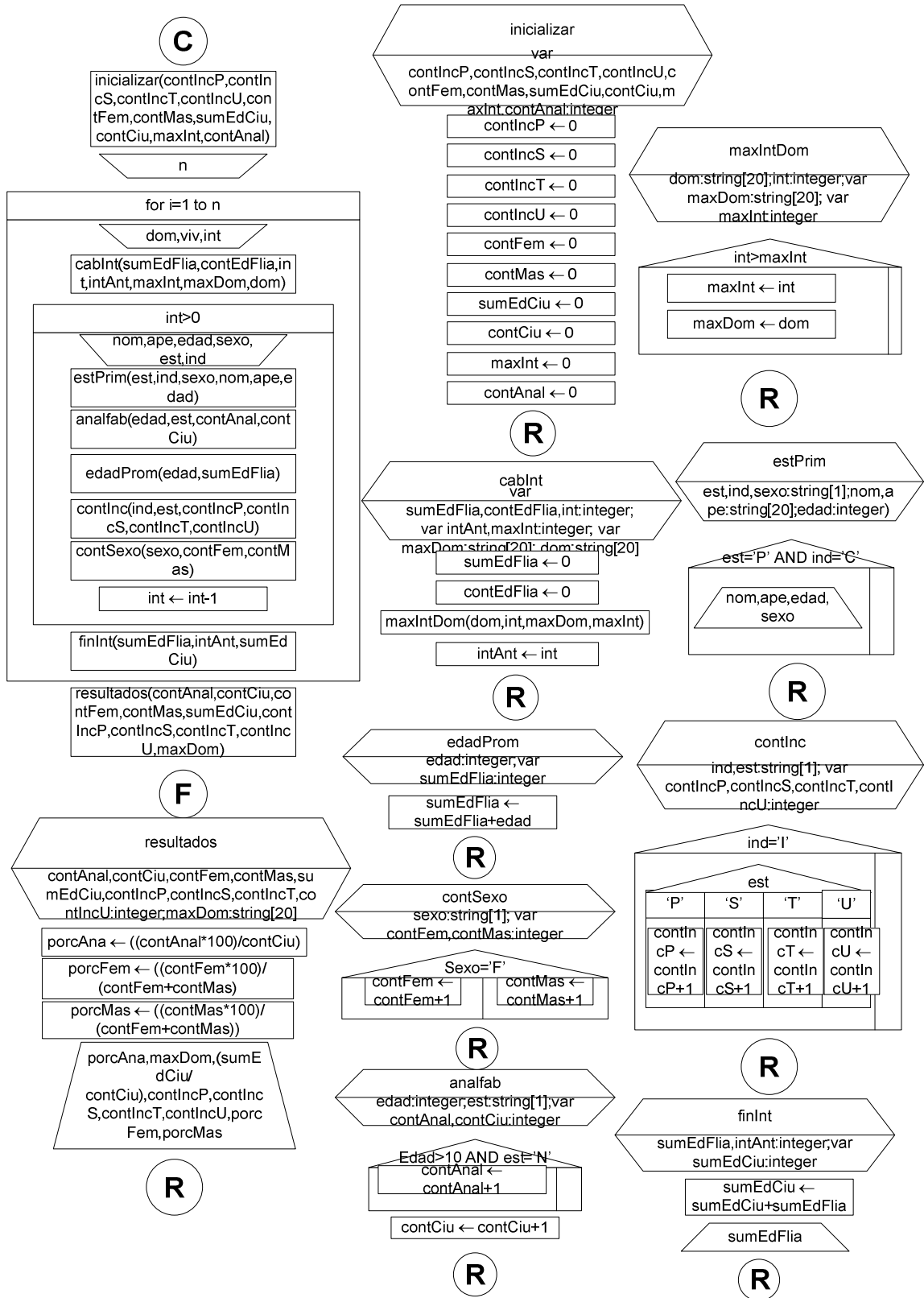
    finInt(sumEdFlia, intAnt, sumEdCiu);

end;

    resultados(contAnal, contCiu, contFem, contMas, sumEdCiu, contIncP, contIncS, contIncT, contIncU, maxDom);

end.

```



Ejercicio N° 35

Una compañía aérea desea emitir un listado con los movimientos mensuales de sus M vuelos al exterior. Para ello cuenta con la siguiente información. De cada vuelo realizado el número de vuelo, destino, y cantidad de asientos. De cada pasajero el número de pasaporte y el importe que abonó por el pasaje en dólares. La información finaliza con un número de pasaporte igual a cero. Se pide emitir el siguiente listado:

Nro. de Vuelo	9999	Destino:	XXXXXXXXXXXXXXXXXX
Nro. de Pasaporte		Importe en u\$s	
99999999		999.99	
99999999		999.99	
Total recaudado del vuelo:		99999.99	
% de Asientos Libres del vuelo		999.99	
% de Asientos Ocupados del vuelo		999.99	
Total recaudado en el mes		999999.99	
Cantidad de veces seguidas que se dieron vuelos completos		99	
El número de vuelo que más recaudó		9999	

```
procedure cabVuelo(var totRecVue:integer;var cantPas:integer);
begin
    totRecVue:=0;
    cantPas:=0;
end;
```

```
procedure proceso(pa:integer;imp:integer;var
totRecVue:integer;var cantPas:integer);
begin
    writeln('Pasaporte: ',pa);
    writeln('importe: ',imp);
    totRecVue:= totRecVue+imp;
    cantPas:= cantPas+1;
end;
```

```
procedure procesarSeguidilla(cantPas:integer;ca:integer; var
                             vueComp:integer;var maxVC:integer);
begin
    if cantPas= ca then begin
```

```

        vueComp:=vueComp+1;
    end else begin
        if vueComp>maxVC then begin
            maxVC:=vueComp;
        end;
        vueComp:=0
    end;
end;

procedure inicializar(var totGral:integer; var
vueMaxRec:integer;var
    vueComp:integer;var maxVC:integer);
begin
    totGral:=0;
    vueMaxRec:=0;
    vueComp:=0;
    maxVC:=0;
end;

procedure finVuelo(totRecVue:integer;cantPas:integer;ca:integer;
var totGral:integer;var vueMaxRec:integer;var nvMaxRec:integer;
nc:integer;var vueComp:integer;var maxVC:integer;nv:integer);
var porcOcupados,porcLibres:real;
begin
    writeln('Recaudacion Vuelo: ',totRecVue);
    porcOcupados:= cantPas*100/ca;
    porcLibres:= 100- porcOcupados;
    writeln('Porcentaje de asientos libres del vuelo:
',porcLibres:0:0);
    writeln('Porcentaje de asientos ocupados del vuelo:
',porcOcupados:0:0);
    totGral:=totGral+totRecVue;
    if totRecVue>vueMaxRec then begin
        vueMaxRec:=totRecVue;
        nvMaxRec:=nv;
    end;
    procesarSeguidilla(cantPas, ca, vueComp, maxVC);
end;

```

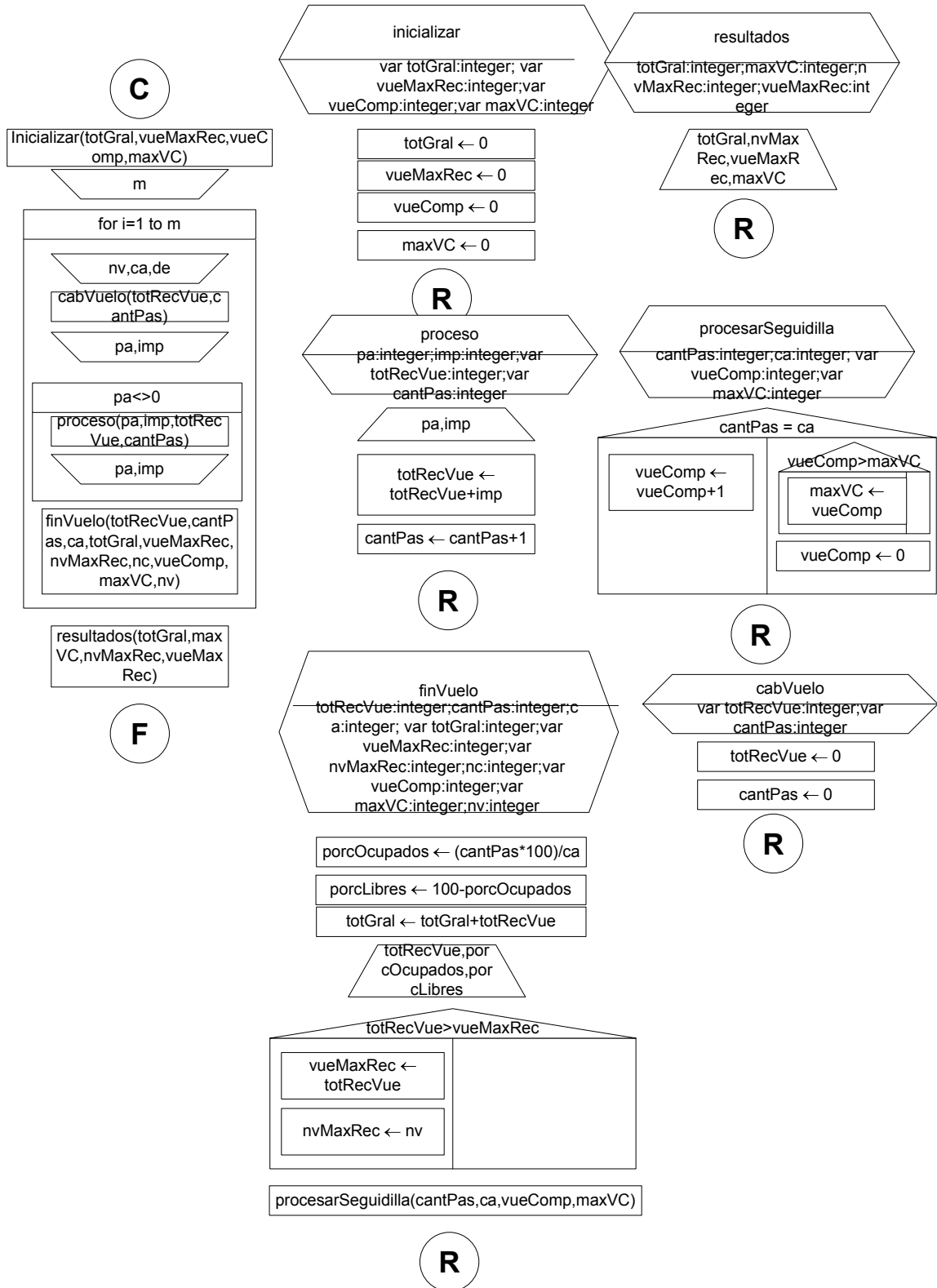
```

procedure
resultados(totGral:integer;maxVC:integer;nvMaxRec:integer;
    vueMaxRec:integer);
begin
    writeln('Total recaudado en el mes: ',totGral);
    writeln('Cantidad de veces seguidas de vuelos completos:
',maxVC);
    writeln('Vuelo de mayor recaudacion: ',nvMaxRec);
end;

//programa principal
var totGral,vueMaxRec,vueComp,maxVC,totRecVue,cantPas,pa,imp,ca,
nvMaxRec,nc,m,i,nv:integer;
var de:string[20];
begin
    inicializar(totGral,vueMaxRec,vueComp,maxVC);
    writeln('Cantidad de vuelos a procesar: ');
    readln(m);
    for i:=1 to m do begin
        writeln('Numero de vuelo: ');
        readln(nv);
        writeln('Cantidad de asientos: ');
        readln(ca);
        writeln('Destino: ');
        readln(de);
        cabVuelo(totRecVue,cantPas);
        writeln('Pasaporte: ');
        readln(pa);
        writeln('Importe: ');
        readln(imp);
        while pa<>0 do begin
            proceso(pa,imp,totRecVue,cantPas);
            writeln('Pasaporte: ');
            readln(pa);
            writeln('Importe: ');
            readln(imp);
        end;
    end;
end;

```

```
        finVuelo(totRecVue,cantPas,ca,totGral,vueMaxRec,nvMaxRec,n  
c,vueComp,maxVC,nv);  
        end;  
        resultados(totGral,maxVC,nvMaxRec,vueMaxRec);  
end.
```

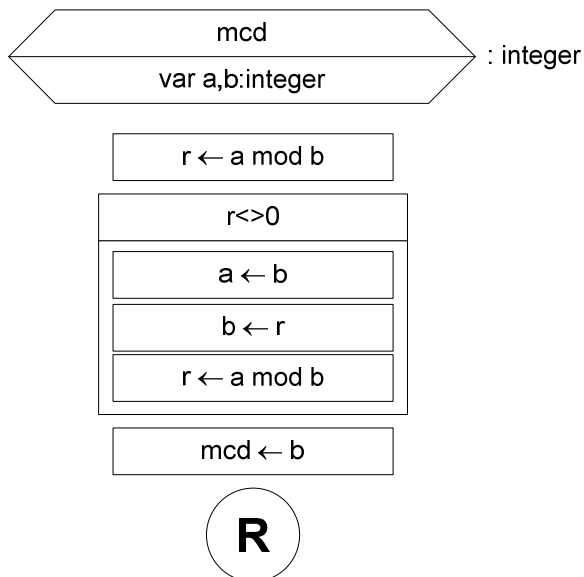


Ejercicio N° 37

Desarrollar una función que calcule el máximo común divisor de dos números enteros A, B con el siguiente algoritmo:

1. Dividir A por B, y calcular el resto ($0 < R < B$)
2. Si $R = 0$, el MCD es B, si no seguir en el punto 3
3. Reemplazar A por B, B por R, y volver al paso 1

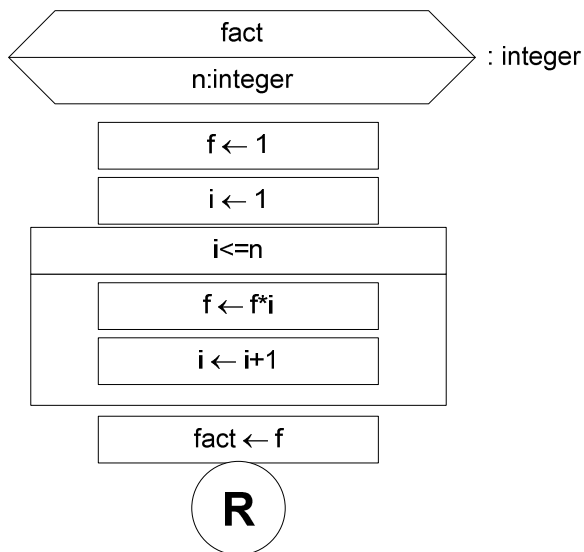
```
function mcd(var a,b:integer):integer;  
var r:integer;  
begin  
    r:=(a mod b);  
    while (r<>0) do begin  
        a:=b;  
        b:=r;  
        r:=(a mod b);  
    end;  
    mcd:=b;  
end;
```



Ejercicio N° 38

Desarrollar una función que dado un número entero positivo calcule y retorne su factorial.

```
var n, fact, i: integer;  
begin  
    writeln('Ingrese un numero para calcular su factorial: ');  
    read(n);  
    fact:=1;  
    i:=1;  
    while (i<=n) do begin  
        fact:=fact*i;  
        i:=i+1;  
    end;  
    writeln('El factorial es: ', fact);  
end.
```



Ejercicio N° 40

Dada una serie de números enteros, informar:

1. su factorial
2. cuantos múltiplos de 3
3. cuantos múltiplos de 5
4. cuantos múltiplos de 3 y de 5

Utilice las funciones de los ejercicios MI-38 y MI-39.

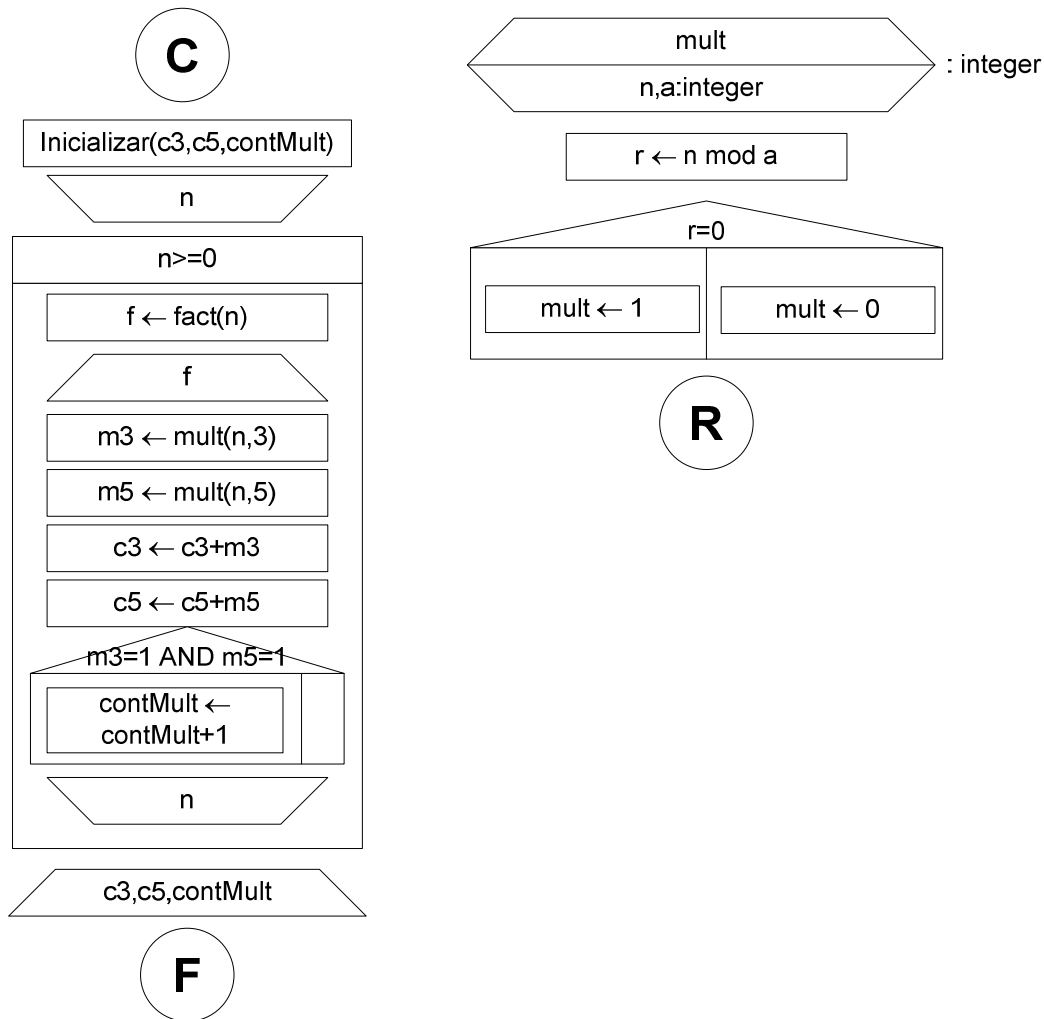
```
procedure inicializar(var c3,c5,contMult:integer);
begin
    c3:=0;
    c5:=0;
    contMult:=0
end;
```

```
function fact(n:integer):integer;
var f,i:integer;
begin
    f:=1;
    i:=1;
    while (i<=n) do begin
        f:=f*i;
        i:=i+1;
    end;
    fact:=f;
end;
```

```
function mult(n,a:integer):integer;
var r:integer;
begin
    r:=n mod a;
    if (r=0) then begin
        mult:=1;
    end else begin
        mult:=0;
    end;
end;
```

```
//programa principal
var f,n,m3,m5,c3,c5,contMult:integer;
begin
    inicializar(c3,c5,contMult);
    writeln('Ingrese un numero: ');
    read(n);
    while (n>=0) do begin
```

```
        f:=fact(n);
        writeln('Su factorial es: ',f);
        m3:=mult(n,3);
        m5:=mult(n,5);
        c3:=c3+m3;
        c5:=c5+m5;
        if ((m3=1) and (m5=1)) then begin
            contMult:=contMult+1;
        end;
        writeln('Ingrese un numero: ');
        read(n);
    end;
    writeln('Cantidad de multiplos de 3: ',c3);
    writeln('Cantidad de multiplos de 5: ',c5);
    writeln('Cantidad de multiplos de 3 y 5: ',contMult);
end.
```



Ejercicio N° 41

Dada la fracción P/Q , para P y Q naturales informar la mayor cantidad de simplificaciones. Desarrolle y utilice una función que reciba dos números naturales y retorne el menor factor común. Ej: $360/60 = 180/30 = 90/15 = 30/5 = 6/1$

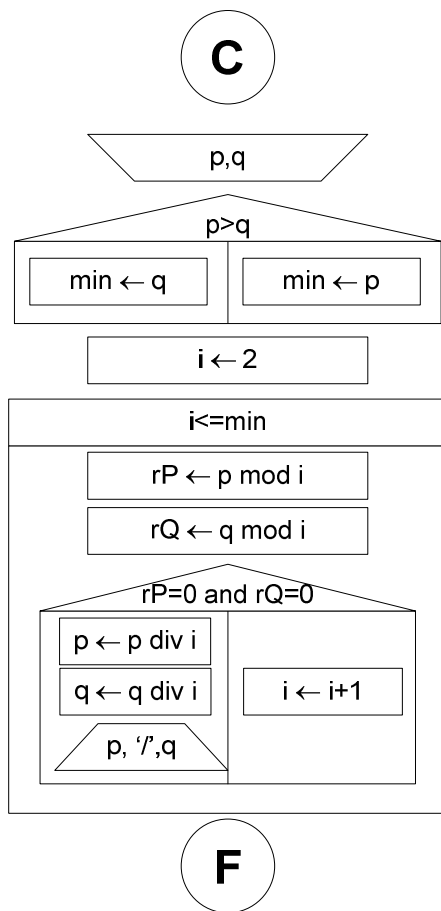
```

var p,q,min,i,rP,rQ:integer;
begin
  writeln('Ingrese el numerador: ');
  read(p);
  writeln('Ingrese el denominador: ');
  read(q);
  if (p>q) then begin
    min:=q;
  
```

```

end else begin
    min:=p;
end;
i:=2;
while (i<=min) do begin
    rP:= p mod i;
    rQ:= q mod i;
    if ((rP=0) and (rQ=0)) then begin
        p:=p div i;
        q:=q div i;
        writeln(p, '/', q);
    end else begin
        i:=i+1;
    end;
end;
end.

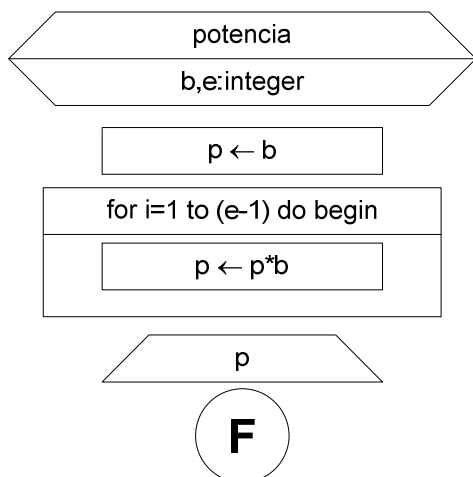
```



Ejercicio N° 43

Desarrollar un procedimiento que dado una base y un exponente, enteros positivos, calcule y retorne la potencia.

```
procedure potencia(b,e:integer);  
var p:integer;  
begin  
    p:=b;  
    for i:=1 to (e-1) do begin  
        p:=p*b;  
    end;  
    writeln(p);  
end;
```



Ejercicio N° 45

Desarrollar un procedimiento que dada una fecha (AAAAMMDD) y un número natural que representa una cantidad de días, calcule y retorne la nueva fecha en tres parámetros año (AAAA), mes (MM) y día (DD) que resulte de incrementar al parámetro fecha con el parámetro cantidad de días.

```
function febrero(a:integer):integer;  
var i,x:integer;  
begin  
    i:=0;  
    repeat  
        x:=4*i;
```

```

        if (x=a) then begin
            febrero:=29;
        end else begin
            i:=i+1;
            if (x>a) then begin
                febrero:=28;
            end;
        end;
    until (x>=a);
end;

function getAnio(fec:longint):integer;
begin
    getAnio:=fec div 10000;
end;

function getMes(fec:longint;a,m:integer):integer;
var x:integer;
begin
    x:=fec mod 10000;
    getMes:=x div 100;
end;

function getDia(fec:longint):integer;
begin
    getDia:=fec mod 100;
end;

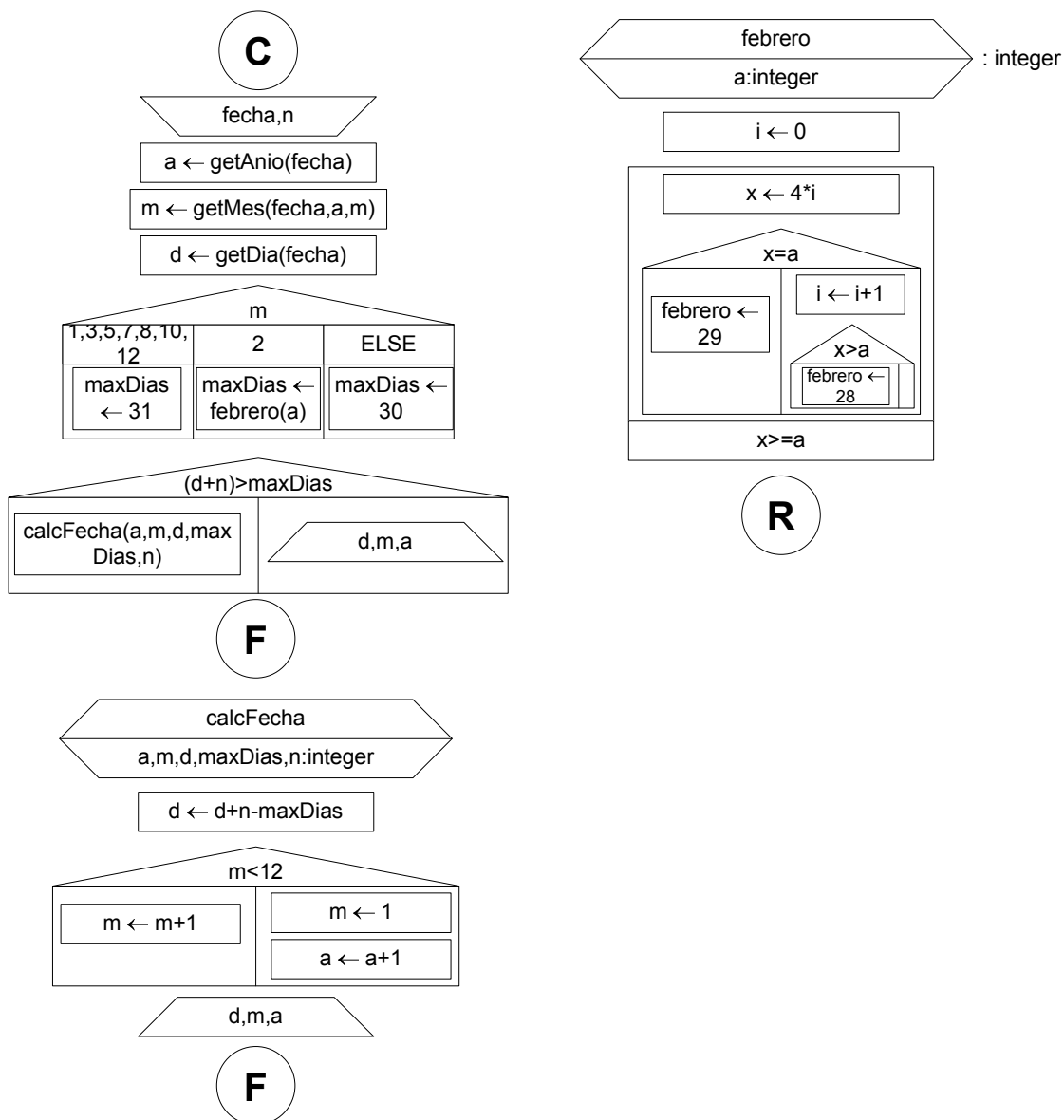
procedure calcFecha(a,m,d,maxDias,n:integer);
begin
    d:=(d+n)-maxDias;
    if (m<12) then begin
        m:=m+1;
    end else begin
        m:=1;
        a:=a+1;
    end;
    writeln('La nueva fecha es: ', d,'/',m,'/',a);
end;

```

```

var fecha:longint; n,a,m,d,maxDias:integer;
begin
    writeln('Ingrese la fecha (AAAAMMDD): ');
    read(fecha);
    writeln('Ingrese la cantidad de dias que incrementen la
fecha: ');
    read(n);
    a:=getAnio(fecha);
    m:=getMes(fecha,a,m);
    d:=getDia(fecha);
    case m of
        1,3,5,7,8,10,12 : begin maxDias:=31; end;
        2 : begin maxDias:=febrero(a);end;
        else begin maxDias:=30; end;
    end;
    if ((d+n)>maxDias) then begin
        calcFecha(a,m,d,maxDias,n);
    end else begin
        writeln('La nueva fecha es: ', d,'/',m,'/',a);
    end;
end.

```



Ejercicio N° 47

Desarrollar:

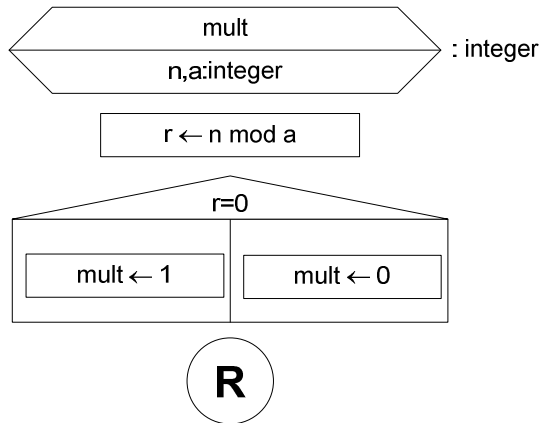
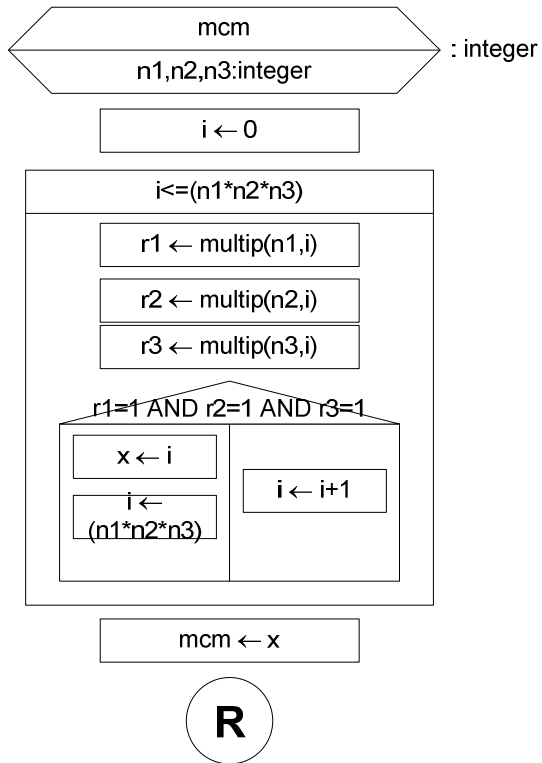
1. Una función que dados tres números naturales (x, y, z) retorne el mínimo común múltiplo.
2. Un procedimiento que dado un valor entero positivo retorne el cociente y el resto de la división entera por 24.

Desarrolle un algoritmo que resuelva la siguiente situación problemática: En una fábrica la sirena amarilla suena cada X horas, la sirena verde cada Y horas y la sirena

azul cada Z horas. Si hoy a las H horas sonaron las tres sirenas a la vez, informar cuantos días y horas deben transcurrir como mínimo para que las tres vuelvan a sonar juntas. Además informar el día mes y año y hora en que se producirá la ocurrencia. Utilice los subprogramas desarrollados en los puntos a), b), el procedimiento del Ejercicio N° 45, e invoque al procedimiento de Pascal Getdate.

1-

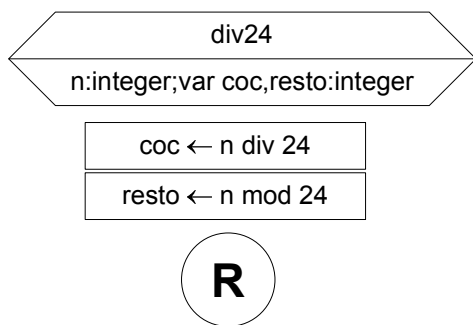
```
function mcm(n1,n2,n3:integer):integer;
var i,r1,r2,r3,x:integer;
begin
    i:=0;
    while (i<=(n1*n2*n3)) do begin
        r1:=multip(n1,i);
        r2:=multip(n2,i);
        r3:=multip(n3,i);
        if ((r1=1) and (r2=1) and (r3=1)) then begin
            x:=i;
            i:=(n1*n2*n3);
        end else begin
            i:=i+1;
        end;
    end;
    mcm:=x;
end;
```



2-

```

procedure div24(n:integer; var coc,resto:integer);
begin
    coc:=n div 24;
    resto:=n mod 24;
end;
  
```



Ejercicio N° 48

Desarrollar:

1. Una función que reciba un número natural de 4 dígitos en formato de HHMM que corresponde a un tiempo determinado en horas y minutos, retorne ese tiempo en minutos.
2. Un procedimiento que reciba el costo en pesos de un abono telefónico, la cantidad de minutos libres que incluye el abono, el cargo en pesos por minuto excedente y la cantidad de minutos utilizados por un abonado, retorne la cantidad de minutos excedidos y el monto en pesos a abonar (costo del abono más minutos excedidos por el costo de minutos excedidos) más el 21% del valor del IVA

Desarrolle un algoritmo que resuelva la siguiente situación problemática: Todos los fines de mes, una empresa de telefonía celular debe confeccionar las facturas con los consumos de todos sus abonados, que se realizan en tres turnos de trabajo: Mañana, tarde y Noche.

Para ello se ingresarán por teclado la siguiente información de cada celular:

- Número de celular 9 dígitos (0 indica cambio de turno)
- Nombre del abonado 20 caracteres
- Dirección del abonado 25 caracteres
- Tiempo utilizado 4 dígitos en formato HHMM
- Tipo de abono (carácter, A, B, C, D o E)

Dependiendo del tipo de abono que se tenga, el usuario tiene cierta cantidad de minutos libres, por los cuales no abona cargo extra, pero por cada minuto que se exceda debe abonar una suma extra según la siguiente tabla:

Plan	A	B	C	D	E
Costo	\$70	\$55	\$40	\$28	\$19
Minutos Libres	300	200	100	60	40
Cargo por minuto excedente	\$0.09	\$0.15	\$0.21	\$0.29	\$0.37

Emitir el siguiente listado por cada turno:

Turno Mañana				
Nombre del Abonado	Dirección	Minutos libres	Minutos Excedidos	Monto Total a Abonar
XXXXXXXXXX	XXXXXXXX	99999	99999	\$ 9999.99
XXXXXXXXXX	XXXXXXXX	99999	99999	\$ 9999.99
XXXXXXXXXX	XXXXXXXX	99999	99999	\$ 9999.99
.....				
Turno Tarde				
.....				
Turno Noche				
.....				

Informar por cada turno de trabajo:

- El nombre del abonado que debe abonar la factura más costosa y el monto de esta.
- El nombre y el número de celular de la persona que realizó la mayor cantidad de minutos excedentes, indicando cuantos fueron.

Al final del día informar:

- El monto total facturado
- En que turno y en que orden dentro del turno apareció el abonado que utilizó la menor cantidad de minutos en el mes.

Utilice los subprogramas a) y b).

```
procedure inicializar(var numTurno:integer;var totalFact:real);
```

```
begin
```

```
    numTurno:=0;
```

```
    totalFact:=0;
```

```
end;
```

```
function minutos (min:longint):integer;
```

```
var hh,mm:integer;
```

```
begin
```

```
    hh:= min div 100;
```

```

        mm:= min - (hh*100);
        minutos:= (hh*60) + mm;
end;

procedure cabFact(var maxPag:real;var PosTurno:integer;var
minExMax:integer);
begin
    maxPag:=0;
    PosTurno:=0;
    minExMax:=0;
end;

procedure calcPago(costoAb:integer;
    minLib:integer;
    costMinEx:real;
    var min:integer;
    var minEx:integer;
    var pagar:real);
    var subtotal:real;
    var minExpagar:real;
begin
    minEx:=min - minLib;
    if minEx>0 then begin
        minExpagar:=minEx*costMinEx;
    end else begin
        minExpagar:=0;
        minEx:=0;
    end
end

```

```
    end;  
    subtotal:=costoAb+minExPagar;  
    pagar:=subtotal + (subtotal*21/100);  
end;
```

```
procedure proceso(var NumTurno:integer;  
    tipo:string[20];  
    hhmm:longint;  
    min:integer;  
    num:longint;  
    nom:string[20];  
    dir:string[20];  
    var pagar:real;  
    var minEx:integer;  
    var minLib:integer;  
    var maxPag:real;  
    var maxPagNom:string[20];  
    var minExMax:integer;  
    var minExMaxNom:string[20];  
    var minExMaxNum:longint;  
    var minExMin:longint;  
    var minExMinNom:string[20];  
    var minExMinPos:integer;  
    var totalFact:real;  
    NumTurnoMin:integer;  
    var PosTurno:integer);  
    var costoAb:integer;
```

```
    var costMinEx:real;
begin
    PosTurno:=PosTurno+1;
    if tipo='a' then begin
        costoAb:=70;
        minLib:=300;
        costMinEx:=0.09;
    end;
    if tipo='b' then begin
        costoAb:=55;
        minLib:=200;
        costMinEx:=0.15;
    end;
    if tipo='c' then begin
        costoAb:=40;
        minLib:=100;
        costMinEx:=0.21;
    end;
    if tipo='d' then begin
        costoAb:=28;
        minLib:=60;
        costMinEx:=0.29;
    end;
    if tipo='e' then begin
        costoAb:=19;
        minLib:=40;
        costMinEx:=0.37;
```

```

end;
min:=minutos(hhmm);
calcPago(costoAb,minLib,costMinEx,min,minEx,pagar);
writeln('numero: ',num);
writeln('nombre: ',nom);
writeln('Direccion: ',dir);
writeln('Minutos Libres: ',minLib);
writeln('Minutos Excedentes: ',minEx);
writeln('Total a pagar: ',pagar:0:2);
if pagar>maxPag then begin
    maxPag:=pagar;
    maxPagNom:=nom;
end;
if minEx>minExMax then begin
    minExMax:=minEx;
    minExMaxNom:=nom;
    minExMaxNum:=num;
end;
totalFact:=totalFact+pagar;
if minEx<minExMin then begin
    minExmin:=minEx;
    minExminNom:=nom;
    minExminPos:=posTurno;
    NumTurnoMin:=numTurno;
end;
end;

```

```

procedure finFact(var
numTurno:integer;maxPagNom:string[20];maxPag:real;
    minExMaxNom:string[20];minExmaxNum:longint;minExMax:integer);
begin
    numTurno:=numTurno+1;
    writeln('Abonado que pago la factura mas costosa:
',maxPagNom);
    writeln('Importe: ',maxPag:0:2);
    writeln('-----');
    writeln('Abonado que utilizo mas minutos excedentes:
',minExMaxNom);
    writeln('Numero: ',minExMaxNum);
    writeln('Cantidad de minutos: ',minExMax);
end;

procedure resultados(totalFact:real;minExminNom:string[20];
    numTurnoMin:longint;minExminPos:integer);
begin
    writeln('Total Facturado: ',totalFact:0:2);
    writeln('Abonado que utilizo la menor cantidad de minutos:
',minExminNom);
    writeln('Turno: ',NumTurnoMin);
    writeln('Posicion: ',minExminPos);
end;

//programa principal
var pagar,totalFact,maxPag:real;
var numTurno,PosTurno,minExMax,minEx,minLib,minExminPos,

```

```

numTurnoMin,min:integer;
var num,hhmm,minExMaxNum,minExmin:longint;
var nom,dir,tipo,maxPagNom,minExMaxNom,minExminNom:string[20];
begin
    inicializar(numTurno,totalFact);
    writeln('Escriba un numero de celular: ');
    readln(num);
    while num>=0 do begin
        writeln('Nombre: ');
        readln(nom);
        writeln('Direccion: ');
        readln(dir);
        writeln('Minutos utilizados: ');
        readln(hhmm);
        writeln('Tipo de abono: ');
        readln(tipo);
        cabFact(maxPag,Posturno,minExMax);
        while num<>0 do begin
            proceso(NumTurno,tipo,hhmm,min,num,nom,dir,

pagar,minEx,minLib,maxPag,maxPagNom,minExMax,minExMaxNom,

minExMaxNum,minExMin,minExMinNom,minExMinPos,totalFact,

            NumTurnoMin,Posturno);
            writeln('Escriba un numero de celular: ');
            readln(num);
            writeln('Nombre: ');
            readln(nom);

```

```
        writeln('Direccion: ');
        readln(dir);
        writeln('Minutos utilizados: ');
        readln(hhmm);
        writeln('Tipo de abono: ');
        readln(tipo);
    end;

    finFact(numTurno,maxPagNom,maxPag,minExMaxNom,minExmaxNum,min
ExMax);

        writeln('Escriba un numero de celular: ');
        readln(num);
    end;

    resultados(totalFact,minExminNom,numTurnoMin,minExminPos);
```

end.

