

Práctico 4 - Repetición for

Programación 1

InCo - Facultad de Ingeniería, Udelar

1. Indique qué se exhibirá en la salida estándar al ejecutar cada uno de los siguientes programas. Después, verifique compilando y ejecutando.

(a)

```
program Ejercicio1a;
var aux, n : Integer;
begin
  aux := 2;
  for n := 1 to 4 do
  begin
    aux := aux * n;
    writeln(n, aux)
  end
end.
```

Compile y ejecute el programa para verificar la solución.

(b)

```
program Ejercicio1b;
var a, b : Integer;
begin
  for b := 1 to 3 do
  begin
    if b <= 1 then
      a := b - 1;
    if b <= 2 then
      a := a - 1
    else
      a := a + 1
    end;
    writeln(a)
  end.
```

Compile y ejecute el programa para verificar la solución.

(c)

```
program Ejercicio1c;
var k, bajo : Integer;
begin
  bajo := 1;
  for k := bajo to 3 do
  begin
    bajo := bajo + 2;
    writeln(k, bajo)
  end.
```

```
end
end.
```

Compile y ejecute el programa para verificar la solución.

```
(d) program Ejercicio1d;
var k, alto : Integer;
begin
  alto := 4;
  for k := alto downto 3 do
    write(k, alto)
  end.
end.
```

Compile y ejecute el programa para verificar la solución.

```
(e) program Ejercicio1e;
var i, j, num : Integer;
begin
  num := 1;
  for i := 1 to 3 do
    begin
      num := num + i;
      for j := 1 to num do
        write(j);
      writeln(i)
    end;
  end.
end.
```

Compile y ejecute el programa para verificar la solución.

2. Determine cuáles de los siguientes fragmentos de código producirán la misma salida al ejecutarlos. Suponga que todas las variables son enteras.

- (a)

```
for i := 1 to 3 do
  for j := i+1 to 3 do
    write(i, j)
```
- (b)

```
for i := 1 to 3 do
  write(i, i+1)
```
- (c)

```
for i := 1 to 4 do
  if (i = 1) or (i = 4) then
    write (i)
  else
    write(i,i)
```

Los fragmentos de código que producirán la misma salida al ejecutarlos son el (b) y el (c)

3. Indique el valor final de la variable `sum` al finalizar la ejecución de cada uno de los siguientes fragmentos de código. Suponga que todas las variables son enteras.

```
(a) sum := 0;
    j := 10;
    for ind := 1 to 5 do
    begin
        sum := sum + 2 * ind + 1 + j;
        j := j - 4;
    end;
```

El valor final de sum es 45

```
(b) const TOTAL = 4;
    ....
    sum := 0;
    for ind := 1 to TOTAL do
        for j := 1 to ind do
            sum := sum + ind + j;
```

El valor final de sum es 50

4. Escriba un programa en Pascal que lea de la entrada estándar tres números naturales a , b y n . Se supone n mayor que 0 y a menor o igual que b . El programa debe desplegar todos los múltiplos de n que haya entre a y b .

Ejemplo

```
Ingrese a b y n : 3 17 4
4 8 12 16
```

```
program Pr4Ej4;
var
    a, b, n, i: integer;
begin
    write('Ingrese a b y n : ')
    readln(a, b, n);

    for i := a to b do
        if i mod n = 0 then
            write(i:1, ' ')
    writeln
end.
```

5. Escriba un programa en Pascal que lea de la entrada estándar un número natural n y despliegue en pantalla todos los divisores naturales de n .

Ejemplo

```
n = 116
1 2 4 29 58 116
```

```

program Pr4Ej5;
var
  n, i : integer;
begin
  write('Ingrese un valor para n: ');
  readln(n);

  write(1, ' '); {1 siempre va al principio}

  { busco divisores entre 2 y n/2 }
  for i := 2 to (n div 2) do
    if n mod i = 0 then
      write(i, ' ');

  writeln(n) {n siempre va al final}
end.

```

6. Escriba un programa en Pascal que lea de la entrada estándar un número natural n . A continuación, el programa deberá leer n enteros y luego desplegar en pantalla el mayor y el menor de ellos. Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo

```

Ingrese un valor para n: 8
Ingrese 8 enteros: 5 12 36 4 21 95 12 18
El mayor entero ingresado es: 95
El menor entero ingresado es: 4

```

```

program Pr4Ej6;
var
  maximo ,
  minimo ,
  numero ,
  i, cantidad : integer;
begin
  write('Ingrese un valor para n: ');
  readln(cantidad);
  write('Ingrese ', cantidad, ' enteros: ');

  { inicializacion de maximo y minimo
  con el primer numero de la entrada
  }
  read(numero);
  maximo:= numero;
  minimo:= numero;

```

```

{ siguientes numeros }
for i:= 1 to cantidad - 1 do
begin
  read(numero);
  { actualizar maximo y minimo }
  if numero < minimo then
    minimo:= numero
  else if numero > maximo then
    maximo:= numero
end;

{ mostrar resultados }
writeln('El mayor entero ingresado es: ', maximo);
writeln('El menor entero ingresado es: ', minimo)

end.

```

7. Escriba un programa en Pascal que lea de la entrada estándar *n* enteros positivos, todos menores que 60 y produzca una gráfica de *n* barras horizontales formadas por asteriscos (similar a la que se muestra en el ejemplo). La *k*-ésima barra deberá tener tantos asteriscos como indique el *k*-ésimo entero (de entre los *n* enteros ingresados). Su programa no necesita controlar que los enteros ingresados sean menores que 60 (asuma que así será). Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo de entrada:

Ejemplo
<pre> Ingrese un valor para n: 5 Ingrese 5 enteros positivos: 7 12 17 35 8 ***** ***** ***** ***** ***** </pre>

```

program Pr4Ej7;
const
  SIMBOLO = '*';
var
  n, num, i, j: integer;
begin
  write('Ingrese un valor para n: ');
  readln(n);

  Write('Ingrese ', n:1, ' enteros positivos: ');
  for i:= 1 to n do

```

```

begin
  read(num);
  { barra de num asteriscos }
  for j:= 1 to num do
    write(SIMBOLO);
  writeln
end
end.

```

8. Escriba un programa en Pascal que lea de la entrada estándar un carácter *c* y un natural *n*. El programa debe desplegar un triángulo de *n* líneas formado por el carácter *c* (similar al que se muestra en el ejemplo). La primera línea debe tener *n* ocurrencias de *c*. La segunda línea debe tener *n-1* ocurrencias de *c* (y así sucesivamente). La última línea debe tener 1 ocurrencia de *c*. Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo

```

Ingrese un carácter c: &
Ingrese un valor para n: 8
&&&&&&&&
&&&&&&&
&&&&&&
&&&&&
&&&&
&&&&
&&&
&&&
&&
&

```

```

program Pr4Ej8;
var
  n,i,j : integer;
  c : char;
begin
  write('Ingrese un carácter c: ');
  readln(c);
  write('Ingrese un valor para n: ');
  readln(n);
  for i:= n downto 1 do
  begin
    for j:=1 to i do
      write(c);
    writeln
  end
end.

```

9. Escriba un programa en Pascal que lea dos números naturales x , n de la entrada estándar y calcule la potencia de x elevado a la n . Para este ejercicio, solamente se permite utilizar las operaciones aritméticas elementales de Pascal (+, -, *, /, DIV, MOD). Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo

```
Ingrese un valor para x: 3
Ingrese un valor para n: 4
El resultado de 3 elevado a la 4 es: 81
```

```
program pr4Ej9;
var
  x, n, i, pot : integer;
begin
  write('Ingrese un valor para x: ');
  readln(x);
  write('Ingrese un valor para n: ');
  readln(n);

  { calculo de la potencia }
  pot := 1;
  for i:= 1 to n do
    pot := pot*x;

  writeln('El resultado de ', x:1, ' elevado a la ', n:1, ' es: ', pot:1, '.')
end.
```

10. Escriba un programa en Pascal que calcule el factorial de un número natural n (leído de la entrada estándar). Para este ejercicio, solamente se permite utilizar las operaciones aritméticas elementales de Pascal (+, -, *, /, DIV, MOD). Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo

```
Ingrese un valor para n: 4
El factorial de 4 es 24
```

```
program pr4Ej10;
var
  n, fact, i: integer;
begin
  write('Ingrese un valor para n: ');
```

```

readln(n);

{ cálculo del factorial }
fact := 1;
for i:= 2 to n do
    fact := fact * i;

{ mostrar el resultado }
writeln('El factorial de ', n:1, ' es: ', fact:1)
end.

```

11. Considere la función f tal que $f(x) = x^2 - 18x + 5$, donde x es un valor entero en el entorno de m a n , siendo m y n dos enteros tales que $m \leq n$.

Escriba un programa en Pascal que lea los valores para m y n de la entrada estándar y despliegue en la salida estándar el valor máximo de $f(x)$ para x en ese entorno. Incluya mensajes de salida con etiquetas descriptivas para solicitar y/o exhibir los valores.

Ejemplo

```

Ingrese un valor para m: -5
Ingrese un valor para n: 10
El valor máximo para x en el entorno de -5 a 10 es 120

```

```

program pr4Ej11;
var
    m, n, x, fx, fxMax : integer;
begin
    write('Ingrese un valor para m: ');
    readln(m);
    write('Ingrese un valor para n: ');
    readln(n);

    {inicialización con el valor en m}
    fxMax := sqr(m) - 18*m + 5;
    { Notar que la iteración comienza en m+1 }
    for x:= m + 1 to n do
    begin
        fx := sqr(x) - 18*x + 5;
        { Se compara con el máximo corriente y si es mayor se actualiza }
        if (fx > fxMax) then
            fxMax := fx
    end;

    { mostrar resultados }
    writeln('El valor máximo para x en el entorno de ',
            m:1, ' a ', n:1, ' es ', fxMax:1, '.')
end.

```


12. Considere la función f tal que: $f(x, y) = x^2 - 9xy + y^2$ donde x, y son valores enteros, ambos en el entorno de $-n$ a n (n entero positivo). Escriba un programa en Pascal que lea el valor para n de la entrada estándar y despliegue en la salida estándar el valor máximo de $f(x, y)$ para x e y en ese entorno. Incluya mensajes de salida con etiquetas descriptivas para solicitar y exhibir los valores.

Ejemplo

```
Ingrese un valor para n: 4
El valor máximo para x e y en el entorno de -4 a 4 es 176
```

```
program pr4Ej12;
var
  x, y, f, fMax, n : integer;
begin
  write('Ingrese un valor para n: ');
  readln(n);

  { inicialización de fMax con el valor f(0,0) }
  fMax := sqr(0) - 9*0*0 + sqr(0);

  { iteración sobre el dominio dado}
  for x:= -n to n do
    for y:= -n to n do
      begin
        f := sqr(x) - 9*x*y + sqr(y);
        if (f > fMax) then {actualizar el máximo }
          fMax := f
      end;

  { mostrar el resultado }
  writeln('El valor máximo para x e y en el entorno de -',
          n:1, ' a ', n:1, ' es: ', fMax:1, '.');
end.
```