

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.`

(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.`

(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.`

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

```
(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.
```

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)
 end.
- (b) for i := 1 to 3 do
 write(i, i+1)
 end.
- (c) for i := 1 to 4 do
 if (i = 1) or (i = 4) then
 write (i)
 else
 write(i,i)
 end.

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;
- (b) const TOTAL = 4;

 sum := 0;
 for ind := 1 to TOTAL do
 for j := 1 to ind do
 sum := sum + ind + j;
 end.
 end.

4. Escriba un programa en Pascal que lea de la entrada estándar tres números naturales a , b y n . El programa debe exhibir en pantalla todos los múltiplos de n que haya entre a y b .

Ejemplo

```
a = 3, b = 17, n = 4
4 8 12 16
```

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
```

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
```

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

```
Ingrese un valor para n: 5
Ingrese 5 enteros positivos: 7 12 17 35 8
*****
*****
*****
*****
*****
```

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

<pre>Ingrese un carácter c: & Ingrese un valor para n: 8 &&&&&&&& &&&&&&& &&&&&& &&&&& &&&& &&& && & &</pre>

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

<pre>Ingrese un valor para x: 3 Ingrese un valor para n: 4 El resultado de 3 elevado a la 4 es: 81</pre>

Solución:

```
program potencia;
var
  x, n, i, pot : integer;
begin
  { ingreso de datos }
  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;
```

```

{ desplegar datos de salida}
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

```

Solución:

```

program factorial;
var
    n, fact, i: integer;
begin
    { Lectura de datos }
    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

```

Solución:

```
program maxfx;
var
  m, n, x, fx, fxMax : integer;
begin
  { ingreso de datos }
  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
```

Solución:

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

```

{ iniciar fMax}
fMax := -maxint;

{ 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.

```