

Práctico 3 - Selección

Programación 1 InCo - Facultad de Ingeniería, Udelar

1. (a) Indique cuáles de las siguientes expresiones booleanas son verdaderas.

$0 > 1$ ('b' < 'd') and (0 < 1)
 'A' < 'Z' (5 < 1) or (0 > -1)
 TRUE < FALSE

- (b) Suponiendo que $p = \text{TRUE}$, $q = \text{FALSE}$ y $r = \text{TRUE}$, indique cuáles de las siguientes expresiones booleanas son verdaderas.

not p or q and r p and q and (3 < 5)
 q and p or not r not (p and r) or (p and r)
 not (p and q and r)

2. Indique el valor de cada una de las siguientes expresiones para $x = 0$, según el tipo de evaluación utilizado. Escriba true, false o error según corresponda.

Circuito corto	Circuito completo	Expresión
false	error	$(x \neq 0) \text{ and } (\text{abs}(x)/x \neq 1)$
error	error	$(x = 0) \text{ and } (\text{abs}(x)/x \neq 1)$
error	error	$(\text{abs}(x)/x \neq 1) \text{ and } (x \neq 0)$
error	error	$(x \neq 0) \text{ or } (\text{abs}(x)/x \neq 1)$
true	error	$(x = 0) \text{ or } (\text{abs}(x)/x \neq 1)$
error	error	$(\text{abs}(x)/x \neq 1) \text{ or } (x \neq 0)$

3. Considere la siguiente instrucción:

```
if (calif >= 90) or (calif < 60) then
  write ('Extrema')
else
  write ('Media')
```

Indique qué valor se exhibe al ejecutarla, según el valor de calif.

calif	Resultado
90	Extrema
60	Media
0	Extrema
70	Media
-90	Extrema

4. Suponga que a y b son variables enteras. Considere las siguientes instrucciones if anidadas:

```
if a > b then
  if a > c then
    write ('a es el grande')
```

Escriba una sola instrucción if que sea equivalente a esta pero que no contenga otra instrucción if anidada.

```
if (a > b) and (a > c) then
  write ('a es el grande')
```

5. Considere el siguiente fragmento de programa:

```
x := 7;
y := 8;
if x > y then
  x := x + 1
else
  y := y + 1
```

Determine el valor que queda en las variables **x** e **y** tras su ejecución.

```
x = 7, y = 9
```

6. Considere el siguiente programa:

```
program Pr3Ej6;
var x, y, z : Integer;
begin
  x := 10;
  y := 11;
  z := 12;
  if (x > y) or (z > y) then
    if x > z then
      if y > z then
        writeln ('Termine.')
      else writeln ('No termino aun.')
    else writeln ('Nunca llega aqui.')
end.
```

(a) Indique qué se exhibirá en la salida estándar al ejecutarlo. Después, ejecútelo en máquina y compare lo que esperaba con la salida que se exhibe.

```
Compile y ejecute el programa para verificar la solución.
```

(b) Reescriba el programa anterior con una indentación adecuada.

```

program Pr3Ej6;
var
  x,y,z : integer;
begin
  x := 10;
  y := 11;
  z := 12;
  if (x > y) or (z > y) then (* siempre entra en el then *)
    if x > z then
      if y > z then
        writeln ('Termine.')
      else writeln ('No termino aun.')
    else writeln ('Nunca llega aqui.')
  end.

```

- (c) Asumiendo que las expresiones booleanas se evalúan por circuito corto, diga si la expresión $z > y$ es evaluada.

La expresión $z > y$ es evaluada ya que la primera subexpresión del `or` evalúa a `false` y por lo tanto es necesario evaluar la segunda subexpresión para determinar el valor de la expresión $(x > y) \text{ or } (z > y)$.

7. Indique qué se exhibirá en la salida estándar al ejecutar el siguiente fragmento de programa:

```

var letra : Char;
...
letra := 'E';
case letra of
  'A': writeln ('El valor es 1. ');
  'E': writeln ('El valor es 5. ');
  'I': writeln ('El valor es 9. ');
  'O': writeln ('El valor es 15. ');
  'U': writeln ('El valor es 21. ');
end

```

El programa despliega: El valor es 5.

8. (a) Escriba una instrucción `case` que sea equivalente a las siguientes instrucciones `if` anidadas. Suponga que todas las variables son de tipo `integer`.

```

if k = 0 then
  r := r + 1
else
  if k = 1 then
    s := s + 1
  else
    if (k = 2) or (k = 3) or (k = 4) then
      t := t + 2

```

```

case k of
  0      : r := r + 1;
  1      : s := s * 1;
  2,3,4 : t := t + 2;
end

```

- (b) Escriba una instrucción case que sea equivalente a las siguientes instrucciones if anidadas. Suponga que la variable calif es de tipo char.

```

if (calif = 'D') or (calif = 'F') then
  writeln ('Trabajo deficiente.')
else
  if (calif = 'C') or (calif = 'B') then
    writeln ('Buen trabajo.')
  else
    if calif = 'A' then
      writeln ('Trabajo excelente.')

```

```

case calif of
  'D','F' : writeln('Trabajo deficiente. ');
  'C','B' : writeln('Buen trabajo. ');
  'A'     : writeln('Trabajo Excelente. ');
end

```

- (c) Sea el siguiente fragmento de programa:

```

val := 3;
case val of
  1: writeln ('uno');
  2: writeln ('dos')
end;
writeln ('Despues del case');

```

¿Qué sucede en Free Pascal cuando es ejecutado? Suponga que la variable val es de tipo integer.

- Aborta.
- Da un mensaje de error.
- Despliega el mensaje 'Despues del case'.**
- Sucede otra cosa.

9. Escriba un programa en Pascal que lea de la entrada estándar un entero de hasta cuatro cifras y lo despliegue en la salida estándar con un punto separando la cifra de los millares (si la hay) de la cifra de las centenas. Su programa no necesita controlar que el entero ingresado tenga hasta cuatro cifras, asuma que así será. Incluya mensajes de salida con etiquetas descriptivas para el ingreso de datos.

Ejemplos
Ingrese un número decimal de a lo sumo 4 cifras: 1234 1.234
Ingrese un número decimal de a lo sumo 4 cifras: 3004 3.004
Ingrese un número decimal de a lo sumo 4 cifras: 7084 7.084
Ingrese un número decimal de a lo sumo 4 cifras: 567 567
Ingrese un número decimal de a lo sumo 4 cifras: 23 23

```
program Pr3Ej9;
const punto = '.';
var numero, resto : integer;
begin
  write('Ingrese un número decimal de a lo sumo 4 cifras: ');
  readln (numero);
  resto := numero mod 1000;
  if (numero >= 1000) then
  begin
    {cifra de millares con punto}
    write (numero div 1000, punto);

    if (resto < 10) then
      {rellenar con 2 ceros}
      writeln ('00', resto:1)
    else if (resto < 100) then
      {rellenar con 1 cero}
      writeln ('0', resto:2)
    else
      { sin relleno }
      writeln (resto:3)
  end
else
  { no hay millares}
```

```
writeln (numero)
end.
```

10. El costo de enviar por correo un paquete es

- \$77 para paquetes que pesan hasta un kilogramo (inclusive)
- \$77 más \$56 por kilogramo adicional o fracción para paquetes que pesan más de un kilogramo.

Escriba un programa en Pascal que, lea de la entrada estándar el peso de un paquete como un número real de kilogramos y exhiba en la salida estándar dicho peso junto al costo del envío, según se muestra en los siguientes ejemplos. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplos
Ingrese peso del paquete: 0.50 Peso Costo 0.50 \$77
Ingrese peso del paquete: 1.00 Peso Costo 1.00 \$77
Ingrese peso del paquete: 1.01 Peso Costo 1.01 \$133
Ingrese peso del paquete: 3.00 Peso Costo 3.00 \$189
Ingrese peso del paquete: 3.40 Peso Costo 3.40 \$245

```
program Pr3Ej10;
const
  costoBase = 77;
```

```

    costoAdicional = 56;
var
    pesoPaquete : real;
    costoPaquete , pesoAdicional : integer;
begin
    write('Ingrese peso del paquete: ');
    readln(pesoPaquete);
    (* el costo siempre vale al menos costoBase *)
    costoPaquete := costoBase;
    if pesoPaquete > 1 then
    begin
        (* cantidad de kgs adicionales *)
        pesoAdicional := trunc(pesoPaquete - 1);
        if pesoAdicional <> (pesoPaquete - 1) then
            (* si tiene parte fraccionaria suma 1kg extra *)
            pesoAdicional := pesoAdicional + 1;
        costoPaquete := costoPaquete + costoAdicional * pesoAdicional
    end;
    writeln('Peso Costo');
    writeln(pesoPaquete:0:2, ' $',costoPaquete:0)
end.

```

11. Escriba un programa en Pascal que lea de la entrada estándar un entero positivo **n** de exactamente cuatro dígitos y otro entero positivo **d** de exactamente 1 dígito. Si **d** aparece en la representación decimal de **n**, entonces exhiba en la salida estándar el número **n** con un signo de suma (+) debajo de cada aparición de **d**. En caso contrario, exhiba el mensaje 'd no aparece en n'. Su programa no necesita controlar que los enteros ingresados tengan cada uno la cantidad de cifras indicada, asuma que así será. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplos
<pre> Ingrese un entero de 4 digitos: 1100 Ingrese un entero de 1 digito: 0 1100 ++ </pre>
<pre> Ingrese un entero de 4 dígitos: 1100 Ingrese un entero de 1 dígito: 5 5 no aparece en 1100 </pre>

```

program Pr3Ej11;
const
    espacio = ' ';

```

```

    marca    = '+';
var
    numero, digito : integer;
    unidades, decenas,
    centenas, millares : boolean;
begin
    { Ingreso de datos }
    write('Ingrese un entero de 4 digitos: ');
    readln(numero);
    write('Ingrese un entero de 1 digito: ');
    readln(digito);

    { unidades = true si el digito aparece en las unidades de numero
      Análogo para decenas, centenas, millares
    }
    unidades:= numero mod 10 = digito;
    decenas:= numero div 10 mod 10 = digito;
    centenas:= numero div 100 mod 10 = digito;
    millares:= numero div 1000 = digito;

    {Se despliega el número leído}
    writeln(numero:4);
    {Se verifica si ocurre}
    if unidades or decenas or centenas or millares then
    begin
        if millares then write(marca)
        else write(espacio);

        if centenas then write(marca)
        else write(espacio);

        if decenas then write(marca)
        else write(espacio);

        if unidades then write(marca)
        else write(espacio);
        writeln
    end
    else
        writeln(digito:1, ' no aparece en ', numero:4)
    end.

```

12. Escriba un programa en Pascal que encuentre las raíces de la ecuación cuadrática $ax^2 + bx + c = 0$ donde a , b , y c son valores reales leídos de la entrada estándar. El programa deberá imprimir en la salida estándar uno de los siguientes mensajes, según corresponda. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Dos raices reales diferentes: xxx.xx y xxx.xx

Dos raices reales iguales: xxx.xx

Dos raices complejas diferentes: xxx.xx (+/-) i xxx.xx

Ejemplo

Ingrese coeficientes de la ecuación: 2.0 3.0 1.0
Dos raices reales diferentes: -1.00 y -0.50

```
program Pr3Ej12;
var
  a,b,c,x1,x2 : real;
  disc, raizdisc : real;
begin
  write('Ingrese coeficientes de la ecuación: ');
  readln(a,b,c);
  writeln('a = ',a:2:1,' b = ',b:2:1,' c = ',c:2:1);

  disc := sqr(b)-4*a*c;

  if (disc = 0) then {única solución}
  begin
    x1 := -b/(2*a);
    writeln('Dos raices reales iguales : ',x1:6:3)
  end
  else
    if (disc > 0) then {dos raíces reales distintas}
    begin
      raizdisc := sqrt(disc);
      x1 := (-b + raizdisc)/(2*a);
      x2 := (-b - raizdisc)/(2*a);
      writeln('Dos raices reales diferentes: ', x1:6:3, ' y ',
              x2:6:3)
    end
    else {Raices imaginarias}
    begin
      x1 := -b / (2*a); {parte real}
      x2 := sqrt(-disc) / (2*a); {parte compleja}
      writeln('Dos raices imaginarias diferentes: ',x1:6:3,'
              (+/-) i ',x2:6:3)
    end
  end
end.
```

13. Escriba un programa en Pascal que convierta un número positivo hexadecimal de tres dígitos tomado de la entrada estándar en su equivalente en base 10. Un dígito hexadecimal es uno de los dígitos 0 a 9 o A(10), B(11), C(12), D(13), E(14), o F(15). El equivalente decimal de un número hexadecimal de la forma abc se calcula como: $a \times 16^2 + b \times 16 + c$. Su programa deberá leer por la entrada estándar el número hexadecimal y realizar las

operaciones aritméticas necesarias para calcular el número equivalente en base 10. Como resultado, exhiba por la salida estándar el valor hexadecimal ingresado junto con el número equivalente en base 10. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Para este ejercicio se recomienda estudiar el uso de la función `ord` para caracteres en el capítulo 8 del libro, página 334 (`ord` y `chr`).

Ejemplo

Ingrese numero positivo hexadecimal de 3 cifras: 7EB Hexadecimal 7EB = Decimal 2027
--

```
program Pr3Ej13;
const base = 16;
var
    d1,d2,d3,
        resultado : integer;
    c1,c2,c3 : char;
begin
    (* entrada de datos *)
    write('Ingrese numero positivo hexadecimal de 3 cifras: ');
    readln(c1,c2,c3);

    (* calcular primer digito *)
    case c1 of
        (* notar que '0'..'9' es equivalente
           a poner '0','1','2','3','4','5','6','7','8','9' *)
        '0'..'9' : d1:= ord(c1) - ord('0');
        (* análogamente, 'A'..'F' equivale a 'A','B','C','D','E','F' *)
        'A'..'F' : d1:= 10 + ord(c1) - ord('A');
    end;

    (* calcular segundo digito *)
    case c2 of
        '0'..'9' : d2:= ord(c2) - ord('0');
        'A'..'F' : d2:= 10 + ord(c2) - ord('A');
    end;

    (* calcular tercer digito *)
    case c3 of
        '0'..'9' : d3:= ord(c3) - ord('0');
        'A'..'F' : d3:= 10 + ord(c3) - ord('A');
    end;

    (* calcular el valor decimal *)
    resultado := d1 * sqr(base) + d2 * base + d3;
    writeln('Hexadecimal ', c1, c2, c3, ' = Decimal ', resultado:4)
end.
```