

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'` ----> Verdadera, aunque depende del ordenamiento lexicográfico que se utiliza ejemplo ASCII
 ☐ `(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. Indique true, false o error según corresponda.

Circuito corto	Circuito completo	Expresión
FALSE	Error en la ejecución (0 / 0)	$(x \neq 0) \text{ and } (\text{abs}(x)/x \neq 1)$
TRUE	Error en la ejecución	$(x = 0) \text{ and } (\text{abs}(x)/x \neq 1)$
Error en la ejecución	Error en la ejecución	$(\text{abs}(x)/x \neq 1) \text{ and } (x \neq 0)$
Error en la ejecución	Error en la ejecución	$(x \neq 0) \text{ or } (\text{abs}(x)/x \neq 1)$
TRUE OR ind = TRUE	Error en la ejecución	$(x = 0) \text{ or } (\text{abs}(x)/x \neq 1)$
Error en la ejecución	Error en la ejecución	$(\text{abs}(x)/x \neq 1) \text{ or } (x \neq 0)$

el caso de la división de 0 no se puede evaluar, se determina por el operador que presente: NOT, AND, OR

OR es TRUE, si alguna de las partes es verdadera

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')
--> if (a > b) AND (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.

5. Considere el siguiente fragmento de programa :

```

x := 7;
y := 8;
if x > y then --> FALSE
  x := x + 1
else
  y := y + 1  y = 9 y x = 7

```

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

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')
    else writeln ('No termino aun.')
    else writeln ('Nunca llega aqui.')
  end.

```

i) --> Nunca llega aqui.

ii)

```

if (x > y) or (z > y) then // true
  if x > z then // false
    if y > z then
      writeln('termine')
    else
      writeln('No termino aun')
    else // pasa para aca
      writeln('nunca llega aqui')
  // --> Nunca llega aqui.

```

(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.

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

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

7. Indique qué es lo que 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. '); El valor es 5
  'I': writeln ('El valor es 9. ');
  'O': writeln ('El valor es 15. ');
  'U': writeln ('El valor es 21. ');
end

```

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

```

```

program pr3Ej8A;
var
  k, r, s, t: integer;
begin
  r := 0;
  s := 0;
  t := 0;

  readln(k);
  case k of
    0 : r := r + 1;
    1 : s := s + 1;
    2, 3, 4 : t := t + 2;
  end;
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.')

```

```

program pr3Ej8B;
var
  calif : char;
begin
  (* Entrada de datos *)
  write('Ingrese la calificación: ');
  read(calif);

  (* Instrucción *)
  case calif of
    'A' : writeln ('Trabajo excelente!');
    'B', 'C' : writeln ('Buen trabajo. ');
    'D', 'F' : writeln ('Trabajo deficiente. ');
  end;
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 se ejecuta dicho fragmento de programa? 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 ingrese de la entrada estándar un entero de hasta cuatro cifras y lo despliegue en la salida estándar con un punto de la siguiente manera: 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).

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

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, dado el peso de un paquete como un número real de kilogramos (leído de la entrada estándar), exhiba en la salida estándar el peso junto al costo de enviar el mismo, según se muestra en los siguientes ejemplos. Incluya mensajes de salida con etiquetas descriptivas para los valores exhibidos.

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

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 solicitar los valores para **n** y **d**.

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

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 los valores exhibidos)

Dos raíces reales diferentes: xxx.xx y xxx.xx
Dos raíces reales iguales: xxx.xx
Dos raíces complejas diferentes: xxx.xx (+/-) i xxx.xx

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

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 los valores exhibidos.

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