

Capítulo 3: Estructura Alternativa

1. Tipos Boolean, Char y String
2. Operadores relacionales y lógicos. Expresiones lógicas (condiciones)
3. Estructura de Control Selectiva (Alternativa o de Decisión)
4. Estructura de Control Selectiva Múltiple

1. Otros Tipos de Datos

La siguiente tabla completa los tipos de datos que ofrece Pascal.

SIMPLES (escalares estandar)	REALES	<i>SINGLE</i>	O R D I N A L E
		<i>REAL</i>	
		<i>DOUBLE</i>	
		<i>EXTENDED</i>	
	ENTEROS	<i>SHORTINT</i>	
		<i>BYTE</i>	
		<i>WORD</i>	
		<i>INTEGER</i>	
		<i>LONGINT</i>	
	<i>BOOLEAN</i>		
	<i>CHAR</i>		
STRING			

El tipo boolean toma valores TRUE o FALSE, estos valores no pueden ser ingresado (lectura), si pueden ser visualizados (escritura). En general son el resultado de expresiones lógicas (condiciones)

El tipo char permite almacenar un carácter. A continuación se presenta una parte (la mas utilizada) de la tabla ASCII DOS, para la representación de caracteres, cada uno tiene una representación numérica, por ejemplo el carácter 'A' se codifica internamente con el número 65, el carácter blanco como el 32. Para visualizar un carácter debe mantener presionada la tecla **alt** mientras digita su respectivo código numérico.

ACII	0	1	2	3	4	5	6	7	8	9
30	-		blanco	i	“	#	\$	%	&	‘
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	¿	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~		Ç	ü

Como se ve en la tabla, existe un orden entre los caracteres y es posible compararlos, por ejemplo la expresión 'A' < 'h' es verdadera y la expresión '4' > 'T' es falsa

Notar que es diferente la representación del valor entero 5 (internamente un 5 en binario) del carácter '5' (encerrado entre apóstrofes, internamente un 53 en binario)

Un tipo string (cadena de caracteres que se trata como una unidad de información) es una secuencia de hasta 255 caracteres, delimitada entre apóstrofes.

Es posible limitar el número de caracteres de un string especificando entre corchetes dicha cantidad, por ejemplo string [n], con n entero menor a 255. La longitud exacta de la cadena almacenada se obtiene con la función length(...), dicho valor coincide con la posición del último carácter.

Se puede establecer una relación entre cadenas, ya que al estar conformadas por caracteres el orden de estos determina el resultado de la relación. Por ejemplo 'Papa' > 'PAPA' es verdadero.

Los tipos integer, boolean, char se los conoce como ordinales. Un tipo de datos es ordinal porque el conjunto de valores que representa se puede contar, es decir, se puede establecer una relación uno a uno entre sus elementos y el conjunto de los números naturales. Esta relación es de orden porque se puede establecer un predecesor y un sucesor para cada valor del tipo ordinal. Esta característica será importante en la implementación de ciertas estructuras de control (Case, For).

2. Operadores de caracteres y cadenas, relacionales y lógicos

Cada tipo de dato, tiene sus propios operadores, tomando las siguientes variables:

Z, Y : integer;

E :real;

Car :char;

Cad :string;

OPERADORES		Tipos de operandos	Tipo de resultado	EJEMPLOS
ALFANUMERICO	+	char, string	string	Cad + 'ena' Car + 'A'
RELACIONAL	< , > , <= , >= , = , <>	todos	boolean	2<= E , 'Papa'> Cad Z +Y * 5 <> 0
LOGICO	NOT , AND , OR	boolean	boolean	(2<= E) AND (Cad > 'Pala')

- ✓ Orden de prioridad : aritmético y alfanumérico – lógico – relacional
- ✓ Dicha prioridad se altera utilizando paréntesis
- ✓ Si en una expresión el orden de precedencia de los operadores es el mismo, se resuelve de izquierda a derecha.

Otras funciones provistas por el lenguaje Pascal

Función	Tipo del Argumento	Resultado
Uppcase(x)	char	devuelve la mayúscula del argumento si éste es una letra minúscula, en otro caso devuelve el mismo caracter
Odd(x)	entero	devuelve el valor lógico True si el argumento de la función es impar y False si es par
Random [(n)]	entero (opcional)	devuelve un número aleatorio. Sin argumento el número aleatorio real entre 0 y 1. Con argumento el número aleatorio entero entre 0 y n -1
Length (S)	S: cadena de caracteres	Proporciona la longitud lógica de S



Resolver el ítem 1 de los ejercicios propuestos al final del capítulo.

3. Estructura de control selectiva

Esta estructura nos permite evaluar una condición y decidir cuales sentencias ejecutar, entre dos alternativas excluyentes.

```

if Condición then
    sentencias 1      {se ejecutan si la condición es verdadera}
[ else
    sentencias 2 ]    {se ejecutan si la condición es falsa, puede estar vacía}
  
```

Después de ejecutar la alternativa que corresponde, sigue en secuencia con la sentencia que está a continuación de la estructura selectiva.

No va un ; antes del ELSE, porque estaría indicando fin de la sentencia.

Nota: las sentencias 1 y 2 pueden ser simples o compuestas (encerradas entre begin - end)

Ejemplo1 - Si quiero calcular $f(x) = 1/x$ ¿Qué sucede si ingresa un $x = 0$?

```

Program evalua;
Var
    X : real;
begin
  readln(X);
  if X <> 0 then
    writeln( 1/X:8:2)
  else
    writeln('no se puede evaluar la función con argumento cero');
end.
  
```

Ejemplo2 - Leer un precio, si supera los \$ 100 decrementar en un 15%. Escribir el precio resultante.

```

Program descuento;
Var
    Precio : real;
begin
  readln( Precio );
  if Precio > 100 then
    Precio := Precio*0.85;
  writeln( Precio)
end.
  
```

Aclaración : no van especificaciones de medidas como mts., hs, \$, lts. etc

Ejemplo3-Ingresa dos nros. e informar la distancia entre ambos. Ej : 2,7 → 5 , -2, -8 → 6

```

Program distancia;
Var
    Nro1 , Nro2, Dist : integer;
  
```

```

Begin
readln (Nro1 , Nro2);
if Nro1 > Nro2 then
    Dist := Nro1 - Nro2
else
    Dist := Nro2 - Nro1;
writeln ( Dist);
end.

```

```

Begin
readln (Nro1 , Nro2);
if Nro1 > Nro2 then
    writeln ( Nro1 - Nro2)
else
    writeln (Nro2 - Nro1)
end.

```

```

Begin
readln (Nro1 , Nro2);
Dist := Nro1 - Nro2
if Dist < 0 then
    Dist := Dist * (-1);
writeln ( Dist)
End.

```



Resolver los ítems 2 a 4 de los ejercicios propuestos al final del capítulo.

Ejemplo4 - Leer un número y determinar si es positivo o no lo es

Program determina;

Var

N : integer;

Begín

readln (N);

if N > 0 then

writeln('Positivo')

else

writeln('No es positivo')

end.

Es posible dentro de la alternativa verdadera (then) o falsa (else) volver a evaluar una condición y elegir nuevamente entre dos alternativas excluyentes.

Si además queremos determinar en caso de no ser positivo, si es negativo o cero, debemos seguir analizando en la alternativa falsa dicha situación.

```

if N > 0 then
    writeln('Positivo')
else {negativo o cero}
    if N = 0 then
        writeln('Cero')
    else {por defecto}
        writeln ('Negativo');

```

```

if N > 0 then
    writeln('Positivo')
else {negativo o cero}
    if N < 0 then
        writeln('Negativo')
    else {por defecto}
        writeln ('Cero');

```

Notar que la última alternativa, se dice *por defecto*, ya que es la única posibilidad que resta, por lo tanto no se explicita la condición (sería redundante hacerlo)

A partir de la solución anterior, se presentan 2 propuestas, indicar los resultados obtenidos en cada una de ellas si el valor de N fuera -5. Y si fuera 5?

¿A que conclusión arriba a partir del resultado obtenido?.

```

a) if N>0 then
    writeln ('positivo');
    if N<0 then
        writeln ('Negativo');
    if N = 0 then
        writeln ('Cero');

```

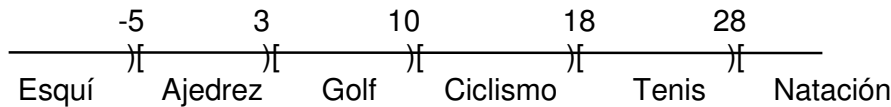
```

b) if N>0 then
    writeln ('positivo');
    if N<0 then
        writeln ('Negativo')
    else
        writeln ('Cero');

```

Ejemplo5 - Dada la siguiente tabla de temperaturas y deportes implementar un algoritmo que lee una temperatura y establezca el correspondiente deporte mostrando el nombre por pantalla

TEMPERATURA < -5° → esquí
 -5° <= TEMPERATURA < 3° → ajedrez
 3° <= TEMPERATURA < 10° → golf
 10° <= TEMPERATURA < 18° → ciclismo
 18° <= TEMPERATURA < 28° → tenis
 28° <= TEMPERATURA → natación



Posibles soluciones:

- ✓ analizar en forma ascendente o descendente los intervalos
- ✓ partir a la mitad, considerando dos grupos, y analizar cada uno de ellos.

Partiendo de la declaración de la variable Temp entera, y ya leída
{se presentan las dos soluciones ¿cuál es más conveniente?}

```

.....
if Temp < -5 then
  writeln ('Esquí')
else
  { Temp >= -5}
  if Temp < 3 then
    writeln ('Ajedrez')
  else
    if Temp < 10 then
      writeln ('Golf')
    else
      if Temp < 18 then
        writeln ('Ciclismo')
      else
        if Temp < 28 then
          writeln ('Tenis')
        else
          writeln ('Natación');
  
```

```

.....
if Temp < 10 then
  if Temp < -5 then
    writeln ('Esquí')
  else
    if Temp < 3 then
      writeln ('Ajedrez')
    else
      writeln ('Golf')
else
  if Temp < 18 then
    writeln ('Ciclismo')
  else
    if Temp < 28 then
      writeln ('Tenis')
    else
      writeln ('Natación');
  
```

Otra solución posible sería evaluar en forma individual cada uno de los intervalos

```

if Temp < -5 then
  writeln ('Esquí');
if (-5<=Temp) and (Temp < 3) then
  writeln ('Ajedrez');
if (3<=Temp) and (Temp < 10) then
  writeln ('Golf');
if (10<=Temp) and (Temp < 18) then
  writeln ('Ciclismo');
if (18<=Temp) and (Temp < 28) then
  writeln ('Tenis');
if (28<=Temp) then
  writeln ('Natación');
  
```

Esta forma requiere más tiempo de ejecución, ya que evalúa todas las opciones y en todas ellas hay que describir el intervalo completo, pues ninguna situación resulta por defecto. Cualquiera de ellas que resulte verdadera no evita la evaluación de las demás

De lo visto anteriormente se resume:

- N estructuras de decisión anidadas permiten n+1 alternativas excluyentes
- N estructuras de decisión secuenciales permiten n alternativas



Resolver los ítems 5 a 8 de los ejercicios propuestos al final del capítulo.

Ejemplo6 - Se quiere calcular el precio que se debe abonar para asegurar un automotor. Los parámetros que determinan el costo son:

- ☞ Tipo de vehículo : C- Comercial (\$200) ; P- Particular (\$ 100)
- ☞ Tipo de seguro : 1- Todo riesgo (+30%) ; 2- Básico (sin incremento)
- ☞ Accidentes en el período anterior : S ; N (– 5%)
- ☞ Edad del conductor (más de 65 años + 10%)

Se parte de un importe inicial (depende del tipo de vehículo) el cual se va incrementando y/o bonificando según los diferentes condicionamientos (tipo de seguro, accidentes, edad).

Program seguro;

Var

Importe : real;

TipoVehic, Accid : char;

TipoSeg, Edad : byte;

Begin

Write('ingrese tipo de vehiculo: C-comercial; P-particular'); readln (TipoVehic);

Write('tuvo accidentes en el periodo anterior: S/N'); readln (Accid);

Write('ingrese tipo de seguro: 1-todo riesgo; 2-basico'); readln (TipoSeg);

Write('ingrese edad'); readln (Edad);

if TipoVehic = 'C' then

Importe :=200

else

Importe :=100;

if TipoSeg = 1 then

Importe := Importe*1.3;

if Accid = 'N' then

Importe := Importe* 0.95;

if Edad >65 then

Importe := Importe*1.1;

Writeln (Importe);

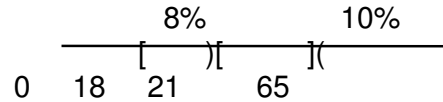
End.

Supongamos que queremos considerar un incremento del 8% en edades entre 18 y 20

- Si es > de 65 años incremento 10 %
- Si es < de 21 años incremento 8 %

Es necesario controlar tres posibilidades diferentes y totalmente excluyentes, con incremento del 10%, del 8% y sin incremento. Encajamos o anidamos las estructuras selectivas

```
if Edad > 65 then
    Importe := Importe * 1.1
else
    if Edad < 21 then
        Importe := Importe * 1.08;
```



Si se cumple Edad > 65 se incrementa el importe en un 10 % y no se evalúa Edad < 21, en cambio si lo planteamos secuencialmente, quedaría :

```
if Edad > 65 then
    Importe := Importe * 1.1;
if Edad < 21 then
    Importe := Importe * 1.08;
```

Si se cumple Edad > 65, después de incrementar el 10 %, igual se evalúa Edad < 21, que en dicho caso sería falso. Por lo tanto conviene encajar o anidar las estructuras selectivas ya que las dos alternativas planteadas son excluyentes y se harían evaluaciones de más

Es importante destacar que no todos las condiciones son dependientes, por ejemplo el tipo de vehículo no incide sobre el tipo de seguro. Estas relaciones deben ser analizadas para determinar si la estructura alternativa es independiente una de otra, o conviene anidarla.



Resolver los ítems 9 a 11 de los ejercicios propuestos al final del capítulo.

Observaciones:

- Si al anidar estructuras selectivas, la estructura interna tiene la alternativa falsa vacía como se muestra a continuación:

<pre>If cond1 then If cond2 then Sentencia a; Else Sentencia b;</pre>	<p><u>Se interpreta</u></p> <pre>If cond1 then If cond2 then Sentencia a; Else Sentencia b;</pre>
---	---

Debe indicarse expresamente que la alternativa **else** pertenece a la estructura externa, delimitando con **Begin-End** la estructura interna, de lo contrario se asocia al **if** más próximo.

Debe expresarse

```
If cond1 then
    begin
        if cond2 then
            Sentencia a;
        end
    Else
        Sentencia b;
```

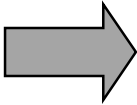
b. Evitar el abuso de IF cuando se usa variables booleanas.

Ejemplo: Sea *Cumple* una variable booleana, X e Y variables del mismo tipo y se desea almacenar True en *Cumple* si ambas almacenen el mismo valor y False en caso contrario. Es más simple sustituir la estructura de decisión, por la asignación a la variable booleana del resultado de la relación que se desea verificar.

```

If  X = Y    then
    Cumple := True
else
    Cumple := False;

```



```

Cumple := X = Y;

```

4. Estructura de decisión Generalizada (o Múltiple)

Cuando en una estructura selectiva anidada, las distintas alternativas dependen del valor resultante de una expresión de tipo ordinal, se puede utilizar la estructura Case

```

Case  expresión of
    Listas de constantes 1 : Sentencias1;
    Listas de constantes 2 : Sentencias2;
    :                      :
    Listas de constantes n : Sentenciasn;
[ else
    SentenciasK; ]
End;

```

El valor que resulta de la expresión es buscado en las listas de constantes, ejecutándose las respectivas sentencias. En caso de no encontrarlo se ejecutan las SentenciasK correspondientes a la alternativa "else". Esta alternativa puede estar vacía (opcional, se expresa entre corchetes)

Ejemplo7 - Leer el número de mes (1..12) e informar el trimestre correspondiente en el año

Program Trimestres;

Var

Mes : byte;

Begin

Readln(Mes);

If Mes <= 3 then

 Writeln ('1er Trimestre')

else

 If Mes <= 6 then

 Writeln ('2do Trimestre')

 else

 If Mes <= 9 then

 Writeln ('3er Trimestre')

 else

 Writeln ('4to Trimestre');

End.

Case Mes of

 1..3 : Escribir ('1er Trimestre');

 4..6: Escribir ('2do Trimestre');

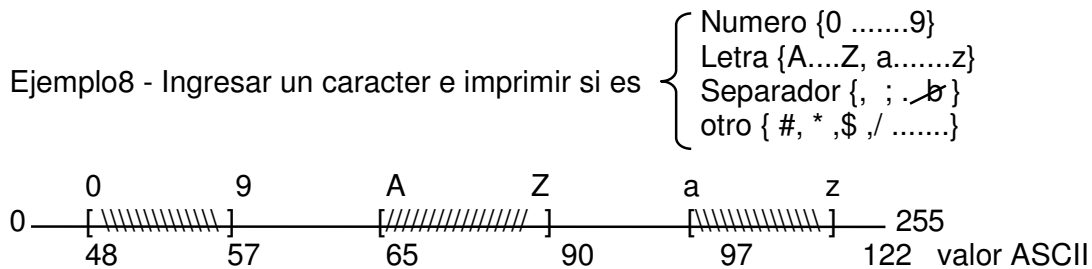
 7..9: Escribir ('3er Trimestre');

 10..12: Escribir ('4to Trimestre');

end;

- ✓ El tipo de la expresión debe ser ordinal (entero, carácter, booleano, subrango (*))
- ✓ Se evalúa el resultado de la expresión y se busca dicho valor en las listas de constantes, si se encuentra se ejecutan las sentencias asociadas. De no encontrarse se ejecuta la opción else o sea la sentenciasK. En ambos casos continua la ejecución en la sentencia que sigue al End
- ✓ Las alternativas son totalmente excluyentes, por lo tanto el mismo valor constante no puede estar en más de una lista.

(*)Subrango: solo para ordinales, por ejemplo : 0..9 ó 'A'..'Z'



Program Caracteres;

Var

Car : char;

Begin

Readln (Car);

If ('0' <= CAR) AND (CAR <= '9') then

writeln('Número')

else

If ('A' <= Car) AND (Car <= 'Z') OR ('a' <= Car) AND (Car <= 'z') then

Writeln('Letra')

else

If (Car = ',') OR (Car = ';') OR (Car = ' ') then

Writeln('Separador')

else

Writeln('Otro');

End.

Utilizando la estructura CASE

.....

Readln(Car)

Case Car of

'0'..'9' : Writeln ('número');

'A'..'Z','a'..'z': Writeln ('Letra');

','',' ',';',' ': Writeln ('Separador');

else

Writeln ('otros');

End;

Ejemplo9 - Leer Dia y Mes, informar a que estación pertenece la fecha leída

Los meses que íntegramente caen dentro de una estación, se agrupan.

La dificultad se presenta en aquellos meses que abarcan dos estaciones distintas, para ello se debe analizar el día de la fecha ingresada. Por lo tanto, estos casos se resuelven aparte.

```
Program Estaciones;
Var
    Dia, Mes : byte;
Begin
    Readln (Dia , Mes);
    Case Mes of
        1 , 2 :   Writeln('verano');
        4 , 5 :   Writeln ('otoño');
        7 , 8 :   Writeln ('invierno');
        10 , 11 : Writeln ( 'primavera');
        3 : If DIA <= 20 then
            Writeln ('verano')
        else
            Writeln ('otoño');
        6 : If DIA <= 20 then
            Writeln ('otoño')
        else
            Writeln ('invierno');
        9 : If DIA <= 20 then
            Writeln ('invierno')
        else
            Writeln ('primavera');
        12 : If DIA <= 20 then
            Writeln ('primavera')
        else
            Writeln ('verano');
    End;
End.
```

Ej10 - Leer dos números N1 y N2 (pueden tomar valor cero o uno). Informar si ambos son 1 ó ambos son 0 ,ó son distintos.

```
Program Numeros;
Var
    N1, N2 : byte;  {precondición : solo toma solo dos valores 0 o 1}
Begin
    Readln (N1 , N2);
    Case N1 + N2 of {el selector del SEGUN pude ser sea una suma}
        0 : Writeln('ambos son cero');
        1 : Writeln ('son distintos');
        2 : Writeln ('ambos son uno');
    End;
End.
```

Ej11 - Leer Dia , Mes y Anio, informar si la fecha es correcta

```
Program FechaOk;
Var
  Dia , Mes , CantDias  : byte;
  Anio: Word;
Begin
  Readln (Dia , Mes , Anio);
  If (1<= Mes) and (Mes <= 12) Then
    Begin
      If (Mes =4) or (Mes =6) or (Mes= 9) or (Mes=11) Then
        CantDias := 30
      else
        If Mes = 2 then
          If (Anio mod 4 = 0) and ( Anio mod 100 <> 0) Then
            CantDias := 29
          else
            CantDias := 28
        else
          CantDias :=31
      If (Dia >= 1) and (Dia <= CantDias) Then
        Writeln( 'fecha correcta')
      else
        Writeln('día erróneo, fecha incorrecta')
    End
  else
    Writeln('mes erróneo, fecha incorrecta');
End.
```

Utilizando la estructura CASE , se reemplazaría el código del recuadro por

```
Case Mes of
  4 , 6 , 9 , 11 : CantDias :=30
  2 : if Anio mod 4 = 0 then
    CantDias := 29
  else
    CantDias :=28
  else
    CantDias := 31
End;
```



Resolver el ítem 12 de los ejercicios propuestos al final del capítulo.

Ejercitación

Estructuras de decisión

Indicaciones: *Los ejercicios 1 y 2 deben resolverse en papel.*

1- Determinar el valor de M , N , O , P , Q, E

VAR

M , N , O , P: Integer;

Cad: string[6];

B, L, T : Boolean;

Begin

M := 2; N := 6; O := 0; Cad:= 'casa';

P:= N * 4;

B := M <= N;

N := N +M;

L:= not B or (P > O);

T := B and (P div 8 = M + 1) and (Cad > 'CASA');

O:= 10;

B := T or L and (O < 0); { Recordar que el OR separa términos }

B := (T or L) and (O < 0);

End.

2- Realizar la prueba de escritorio de cada uno de los siguientes programas:

a- VAR

C1, C2 : char;

Begin

Writeln(Íngresar dos letras');

Readln(C1); Readln(C2);

If C1 > C2 then

writeln(C2)

Else

writeln(C1);

writeln(C2+'...' +C)

End.

<u>Lote de prueba 1:</u> 'c', 'a' <u>Lote de prueba 2:</u> 'E', 'e'
--

b- CONST

P='El punto ';

VAR

r, x, y : real;

Begin

Writeln('Ingresar radio y un punto');

Readln(r,x,y);

If $\sqrt{x} + \sqrt{y} \leq \sqrt{r}$ then

Writeln(p, x:4:2, y:4:2, 'pertenece a la circunferencia de radio ' ,r:4:2)

Else

Writeln(p, x:4:2, y:4:2, 'no pertenece a la circunferencia de radio ' ,r:4:2)';

End.

<u>Lote de prueba 1:</u> 2 , 0 , 1 <u>Lote de prueba 2:</u> 2 , 3 , 2 <u>Lote de prueba 3:</u> 5 , 4 , 3
--

3- Leer a, b y c coeficientes de una ecuación cuadrática (ax^2+bx+c). Determinar si tiene raíces reales o imaginarias ($b^2-4ac < 0$).

4-El costo de un pasaje aéreo depende de la clase (\$1000-turista y \$2000-bussines) y recibe un descuento del 10% si viaja la primera semana del mes. Ingresar la clase (T ó B) y el día (1..31) indicar el importe del pasaje.

5-Leer el numero X , calcular e imprimir el resultado de la función Y definida así:

$$Y = \begin{cases} 1 + (1 + x)^2 & \text{si } X < 0 \\ 0 & \text{si } X = 0 \\ 1 - (1 + x)^2 & \text{si } X > 0 \end{cases}$$

6-Agregar el cálculo de las raíces reales (si las tiene) al ejercicio 3.

7- Sean a, b, c y Máximo cuatro variables numéricas.

a- Cuál es la función del siguiente fragmento de programa? Realizar prueba de escritorio.

```
Readln(a,b,c);
Maximo := a;
if b >= Maximo then
    Maximo := b;
if c >= Maximo then
    Maximo := c;
```

b- Si en la ejecución del programa anterior, a, b y c tienen el mismo valor ¿cuántas asignaciones se efectuarán?

c- Si se cambiaran los >= por el signo >, cuando a, b y c tienen el mismo valor, ¿cuántas asignaciones se efectuarán?

8- Se desea conocer el impuesto a pagar según el total ganado en el año. Leer la ganancia GANUAL y mostrar el impuesto IMPU de acuerdo a la siguiente tabla:

GANUAL < 10000.....	IMPU = 0
10000 <= GANUAL < 15000.....	IMPU = 2% de (GANUAL - 10000)
15000 <= GANUAL	IMPU = 300 + 5% de (GANUAL - 15000)

9- Leer número de libreta y tres notas (de tipo real) correspondientes a una materia, mostrar nro. de libreta y dos carteles indicando si aprobó o no. Para aprobar se necesita un promedio de 5 puntos como mínimo.

Además informar el concepto de la siguiente forma:

- ✓ si obtiene 9 o más le corresponde Sobresaliente,
- ✓ menos de 9 y hasta 8 inclusive el concepto es Distinguido,
- ✓ menos de 8 y hasta 6 es Bueno
- ✓ menos de 6 y hasta 4 es Regular,
- ✓ menos de 4 es Insuficiente.

10- Ingresar tres letras y escribirlas ordenadas alfabéticamente. Resolver este problema de dos formas diferentes utilizando: estructuras de decisión independientes y estructuras de decisión anidadas. ¿Cuál es más eficiente?

11- Sean **p, q** expresiones lógicas; **a, b, c, d** acciones.

a- Para la siguiente estructura de decisión, escribir una estructura equivalente tal que se evalúen proposiciones simples (sin conectivos lógicos)

a1- if p and q then a else b;	a2- if p or q then a else b;
--	---

b- ¿Para qué valores de p y q se ejecutan cada una de las acciones ?

b1- if p then a else if q then b;	b2- d; if q then begin if p then b else c; a; end;
---	--

12- Desarrollar un programa para resolver cada uno de los siguientes problemas (utilizar la estructura de decisión generalizada):

a- Dados dos números calcular las cuatro operaciones aritméticas, utilizar un menú de opciones para ingresar un código que indique la operación a realizar, escribir el resultado.

b- Una empresa de transportes realiza envíos de cargas, se desea calcular y mostrar el importe a pagar.

Los datos que se ingresan son PESO de la carga (número real) y CATEGORIA (dato codificado: 1 - común , 2 - especial , 3 – aéreo).

El precio se calcula a \$25 por Kg para categoría común, \$32.5 por Kg para categoría especial y \$50 por Kg para categoría aérea. Se cobra recargo por sobrepeso: 30% si sobrepasa los 15 Kg, 20% si pesa más de 10 Kg y hasta 15Kg inclusive, 10% más de 5Kg y hasta 10 Kg inclusive.