Problema 6.1.

Codifica un programa PASCAL que permita:

- Definir un tipo enumerado con el siguiente rango: Dom, Lun, Mar, Mie, Jue, Vie, Sap, y Dom
- Recoger un entero comprendido entre 0 y 6 .
- Escribir en pantalla el día de la semana correspondiente.
- Si el día es Sábado o Domingo, escribir el mensaje en pantalla `A seguir durmiendo', si es día laborable, escribir en pantalla el mensaje `Vamos a trabajar'.
- Recoger por teclado el número de horas que se ha trabajado durante cada día de esa semana, desde el Lunes hasta el día introducido por teclado, y mostrar el número total de horas trabajadas.
- Utiliza dos bucles for para escribir las siguientes líneas:

- Lunes Martes Miércoles Jueves Viernes Sábado Domingo
- Domingo Lunes Martes Miércoles Jueves Viernes Sábado

- Define una variable Pulsado de tipo anónimo, con dos valores: on, off. Asigna a Pulsado uno de sus valores válidos, y utiliza la estructura alternativa para escribir el mensaje `Estoy conectado' con valor on o `Estoy desconectado' con el valor off.

```
program Enumerados (output);

uses

crt;

type

tDiasSemana=(Dom,Lun,Mar,Mie,Jue,Vie,Sab);

{ tDS=(Lun, Lan, Lon); L¡nea Err¢nea porque duplica el valor Lun}

{ tVocales=('A','E','I','O','U'); Err¢nea han de ser identificadores}

{ tMeses=(1,2,3,4,5,6,7,8,9,10,11,12); Err¢nea han de ser identificadores}

var

x,

HorasTrabajadas, Total:integer;

Dia,i: tDiasSemana;

{Tipo An¢nimo}

Pulsado:(on,off);

procedure TomaDato(var x:integer);
```

```
begin

repeat

write ('Escribe Numero: ');

readln (x);

until (x>=0) and (x<=7);

end;

function Convierte(x:integer):tDiasSemana;

begin

case x of

0: Convierte:=Dom;

1: Convierte:=Lun;

2: Convierte:=Mar;

3: Convierte:=Mie;

4: Convierte:=Jue;

5: Convierte:=Vie;

6: Convierte:=Sab;

end; {del case}

end;

procedure EscribeDia (x:tDiasSemana);

{Escribe una cadena del d¡a, seg£n sea el valor que recibe}

begin

case x of

Lun: write ('Lunes: ');

Mar: write ('Martes: ');

Mie: write ('Mi rcoles: ');

Jue: write ('Jueves: ');
```

```
Vie: write ('Viernes: ');

Sab: write ('Sabado: ');

Dom: write ('Domingo: ');

end; {del case}

end;

begin {Programa Principa}

{− Sepueden pasar como argumentos y devolverse en funciones

− NO se pueden escribir ni leer directamente }

clrscr;

TomaDato(x);

Dia:=Convierte(x);

EscribeDia(Dia);

{Se pueden usar los operadores relacionales}

if (Dia=Sab) or (Dia=Dom) then

writeln ('Seguimos durmiendo')

else

writeln ('Arriba, a estudiar');

{Se pueden usar en bucles for}

for i:=Lun to Dia do

begin

{writeln ('Escribe el n£mero de horas trabajadas el ',Dia);

La instrucci¢n anterior es erronea, no se pueden escribir }

write('Escribe el n£mero de horas trabajadas el ');

EscribeDia(i);

readln (HorasTrabajadas);

total:=total+HorasTrabajadas;
```

```
end;

writeln ('El Número de horas trabajadas hasta ahora es: ',total);

{Sucesor y Predecesor}

for i:=Dom to Vie do

EscribeDia (succ(i));

EscribeDia(Dom);

writeln;

for i:=Lun to Sab do

EscribeDia (pred(i));

EscribeDia (Sab);

{Valores An¢nimos}

writeln;

Pulsado:=on;

if Pulsado=on then

writeln ('Est  en ON')

else

writeln ('Est  en OFF');

end.
```

Problema 6.2.

Escribe un programa que defina dos tipos tPositivo y tMayuscula, cuyos intervalos válidos sean los enteros positivos y las letras mayúsculas respectivamente.

- Define variables de amos tipos y realiza las siguientes operaciones:
- Asigna valores a las variables
- Aplica operadores de sus tipos bas
- Lee y Escribe por teclado y pantalla respectivamente valores para estas variables.

```
program Subintervalo (input,output);

uses

crt;
```

```
const

max=32767;

type

tPositivo=0..max;

tMayuscula='A'..'Z';

var

x,y,z:tpositivo;

r:real;

c:char;

caracter:tMayuscula;

begin

clrscr;

{Pueden asignarse, y aplicar cualquier operador del tipo anfitrión}

x:=10000; {Dentro de rango. Entero sin signo}

y:=20000;

z:=x+y;

r:=z;

{Pueden escribirse y leerse}

write('Escribe car cter en may£scula: ');

readln(caracter);

writeln (caracter);

writeln(z:7);

writeln(r:10:2);

end.
```

Problema 6.3.

Escribe un programa en PASCAL que acepte un número entero positivo de un byte entre 0 y 255 y muestre por pantalla el número en binario.

```
program EnteroByteEnBinario (input,output);
uses
crt;
type
EnteroByte=0..255;
var
n:EnteroByte;
Procedure TomaDato(var n:EnteroByte);
{Postcondicion: 0<=n<=255 }
begin
repeat
write ('Escribe un positivo entre 0 y 255: ');
readln (n);
until (n>=0) and (n<=255)
end;
Procedure MostrarByteReducir(n:EnteroByte; var res:EnteroByte);
{Muestra el bit del orden marcado por n, y prepara la variable res
para la salida y posible próxima llamada }
{Precondición: n>0
Postcondición: output=res div n y res=res mod n}
begin
write((res div n):1);
res:=res mod n;
end;
Procedure Mostrar(n:EnteroByte);
{Muestra, uno a uno, los bit resultantes de las diviones entre los distintos
```

```
ordenes de unidades }

{Precondición: 0<=n<=255

Postcondición: output= representación binaria de n }

var

p:EnteroByte;

begin

p:=128;

write ('El número ',n:4, ' decimal es: ');

repeat

MostrarBityReducir(p,n);

p:=p div 2

until p<=1;

writeln(n:1,' en binario');

end; {De mostrar}

begin {Programa Principal}

clrscr;

TomaDato(n);

Mostrar(n);

end.
```

Problema 6.4.

Escribir un programa en Pascal que defina un tipo con los caracteres alfabéticos `A'a la `z' y que tenga la siguiente funcionalidad:

- Definir la siguiente funcionalidad para el tipo tAlfabetico
    - Unión (+)
    - Intersección (*)
    - Diferencia (−)
    - Igualdad (=)
    - Desigualdad (<>)
    - Inclusión (<= o >=)
    - Pertenencia (in)

- Añadir la siguiente funcionalidad al programa

    - Escribir en pantalla los elementos del conjunto
    - Devolver la cardinalidad del conjunto
    - Inserta un elemento en un conjunto
    - Elimina un elemento de un conjunto

Prueba el programa con las siguientes órdenes:

- Define tres conjuntos con Mayúsculas, Minúsculas y Vocales de tipo t Afabético.
- Recoge un valor de tipo tAlfabético y escribe en pantalla si es mayúscula o no
- Escribe en pantalla si el conjunto vocales, es subconjunto de mayúsculas
- Construye un nuevo conjunto Union que sea la unión de Vocales y Minúsculas
- Construye un nuevo conjunto Intersección que sea la Intersección de Vocales y Minúsculas
- Construye un nuevo conjunto Intersección que sea la Intersección de Vocales y Mayúsculas
- Construye un nuevo conjunto Diferencia, en el que se hayan eliminado del conjunto Union las minúsculas
- Escribe por pantalla si Diferencia y Vocales son iguales.
- Escribe por pantalla el número de elementos del conjunto Union.
- Define un conjunto Alfabeto igual a mayúscula e inserta en él un carácter introducido por teclado
- Elimina un carácter del Conjunto

Solución 6.4.

```
program Conjuntos (input,output);

uses

crt;

type

{tConjunto1=set of integer; Definici¢n Err¢nea. Fuera de Rango}

tAlfabetico=set of 'A'..'z';

var

Vocales,

Mayusculas,Union,Interseccion,Diferencia,Alfabeto,

Minusculas:tAlfabetico;

c:char;

procedure Elimina (c:char; var conj:tAlfabetico);

{Precondicion: c pertenece a conj}

begin
```

```
Conj:=Conj − [c,'A'];
end;
function Cardinalidad(c:tAlfabetico):integer;
var
cont:integer;
i:char;
begin
cont:=0;
for i:='A' to 'z' do
if i in c then
cont:=cont+1;
cardinalidad:=cont;
end;
procedure Inserta (c:char; var conj:tAlfabetico);
var
aux:tAlfabetico;
begin
aux:=[];
if not (c in conj) then
begin
aux:=aux+[c];
conj:=conj+aux;
end;
end;
procedure Escribe (conj:tAlfabetico);
var
```

```
i:char;
begin
for i:='A' to 'z' do
if i in conj then
write ( i,' ');
end;
procedure TomaDato (var c:char);
begin
write('Introduce caracter: ');
readln(c);
end;
function Pertenece (c:char;conjunto:tAlfabetico):boolean;
begin
Pertenece:=c in conjunto;
end;
function Subconjunto (conj1,conj2:tAlfabetico):boolean;
begin
Subconjunto:=Conj1<=Conj2;
end;
begin
clrscr;
Vocales:= ['A','E','I','O','U'];
Mayusculas:= ['A'..'Z'];
Minusculas:=['a'..'z']; {Minusculas est  vacio son valores no válidos}
TomaDato(c);
```

{Operador (in) Pertenencia: Elemento pertenece }

```
if Pertenece(c,Vocales) then
writeln (c,' es vocal may£scula')
else
writeln (c, ' no es vocal may£scula');
{Operador (<=) Inclusi¢n: Subconjunto, est  incluido }
if Subconjunto(Vocales,Mayusculas) then
writeln ('Vocales mayúsculas es subconjunto de Mayusculas')
else
writeln ('Vocales Mayusculas No es subconjunto de May£sculas');
{Operador (+) Union: Uni¢n de Conjuntos}
writeln ('Operador (+) Union: Uni¢n de Conjuntos');
Union:=Vocales+Minusculas;
Escribe(Union);
{Operador (*) Intersecci¢n: Intersecci¢n de Conjuntos}
writeln;
Interseccion:=Vocales*Minusculas; {Conjunto vac¡o}
Escribe(Interseccion);
writeln;
Interseccion:=Vocales*Mayusculas; {Conjunto Vocales}
Escribe(Interseccion);
{Operador (−) Diferencia: Diferencia de Conjuntos}
writeln;
Diferencia:=Union−Minusculas;
Escribe(Diferencia);
{Operador (=) Igualdad: Igualdad de Conjuntos}
writeln;
```

```
if Diferencia=Vocales then
writeln ('Diferencia=Vocales')
else
writeln ('Diferencia<>Vocales');
```

{Operador (<>) Desigualdad: Desigualdad de Conjuntos}

```
writeln;
if Diferencia<>Vocales then
writeln ('Diferencia<>Vocales')
else
writeln ('Diferencia=Vocales');
```

{Cardinalidad: Devuelve el n£mero de elementos}

```
writeln('Union tiene ',Cardinalidad(Union),' elementos');
{Inserta: Inserta un elemento en un conjuto del mismo tipo.
Si ya est  NO lo repite}
writeln;
Alfabeto:= Mayusculas;
Inserta(c,Alfabeto);
Escribe(Alfabeto);
```

{Elimina: Devuelve el conjunto sin el elemento}

```
writeln;
if Pertenece(c,Alfabeto) then
Elimina(c,Alfabeto);
Escribe(Alfabeto);
end.
```