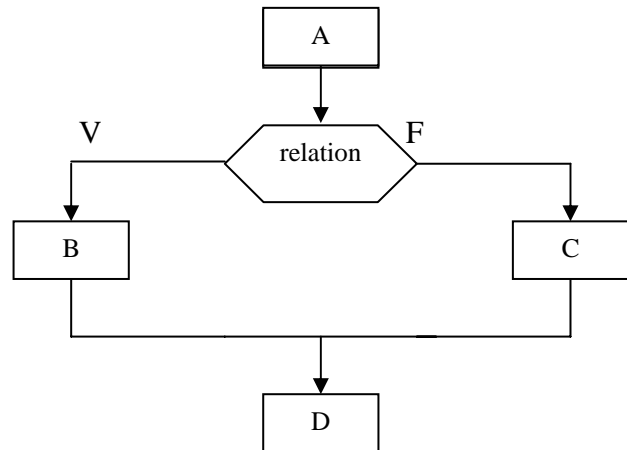


syntaxe :

```
instructions A;
if relation then
  begin
    instructions B ;
  end
else
  begin
    instructions C;
  end;
instructions D;
```

effet :



commentaires :

Si il n'y a qu'une seule instruction B (ou C), le begin end correspondant est inutile.
Le else est facultatif ; il ne doit pas être précédé immédiatement d'un point virgule.

exemple :

```
program second_degre;
var a,b,c,delta,x1,x2:real;
BEGIN

  writeln('écrire a,b,c');
  readln(a,b,c);

  if a = 0 then
    writeln('équ. du 1er degré')
  else
```

commentaires :

déclaration du programme
déclaration des variables
début du programme principal : BEGIN en majuscules
initialisation (par lecture) des variables d'entrée ; writeln est facultatif, readln indispensable
pas de point virgule avant le else

```

begin
  delta:= b*b-4*a*c;
  if delta < 0 then
    writeln('pas de solution');
  if delta = 0 then
    begin
      x1:=-b/2/a;
      writeln(x1);
    end;
  if delta > 0 then
    begin
      x1:=(-b-sqrt(delta))/2/a;
      x2:=(-b+sqrt(delta))/2/a;
      writeln(x1,x2);
    end;
end;
```

ce begin et le else en vis-à-vis délimitent le groupe d'instructions à effectuer si a est non nul
initialisation (par affectation) des variables de sortie

END.