

Algorithmes : Structure alternative : « Si... Alors...Sinon »

Exemple:

Appliquer l'algorithme ci-contre aux nombres 18, 21 et 237.

Exercice : Modifier l'algorithme 4 pour qu'il affiche si un nombre donné est un multiple de 7.

Algorithme 4

```

Si n a pour reste 0 dans la division par 2 Alors
    afficher "nombre pair"
Sinon
    afficher "nombre impair"
FinSi
    
```

La structure alternative s'exprime ainsi :

```

Si condition Alors
    suite d'instructions 1 (si la condition est vraie)
Sinon
    suite d'instructions 2 (si la condition est fausse)
FinSi
    
```

La condition ne doit offrir que deux réponses : vraie ou fausse !
 Quand la condition est vraie, on exécute la suite d'instructions 1 ; quand elle est fausse, on exécute la suite d'instructions 2.


Dans certains cas, il n'y a pas d'instruction à effectuer quand la condition est fausse. On écrit alors :

```

Si condition Alors
    suite d'instructions 1 (si la condition est vraie)
FinSi
    
```

Si la condition est vraie, la suite d'instructions 1 est exécutée. Si la condition est fausse, on passe à la suite de l'algorithme.

La structure alternative de l'algorithme 4 page 12 dans différents langages (programmes complets sur le site) @ @

Algobox	Scratch	TI
<pre> ▼ SI (n%2==0) ALORS DEBUT_SI AFFICHER "le nombre est pair" FIN_SI ▼ SINON DEBUT_SINON AFFICHER "le nombre est impair" FIN_SINON </pre>		<pre> :If ent(N/2)=N/2 :Then :Disp "PAIR" :Else :Disp "IMPAIR" :End </pre>
<pre> Xcas si irem(n,2)==0 alors afficher("nombre pair"); sinon afficher("nombre impair"); fsi;; </pre>	<pre> Scilab 2 if reste(n,2)==0 then 3 disp("nombre pair") 4 else 5 disp("nombre impair") 6 end </pre>	<pre> Casio Graph 35+ If MOD(N,2)=0 Then "PAIR" Else "IMPAIR" IfEnd </pre>

Attention ! Le test d'une égalité s'écrit souvent avec un double signe d'égalité ==

Aide : « $n\%2$ », « $\text{irem}(n, 2)$ », « $\text{reste}(n, 2)$ », « $\text{mod}(n, 2)$ » donnent le reste de la division euclidienne de n par 2.

Les calculatrices TI82, 83 ne disposent pas de cette fonction « reste » : on teste alors si $n/2$ est un entier (en testant s'il est égal à sa partie « avant la virgule » $\text{ent}(n/2)$). Il faut s'adapter au langage de programmation utilisé ...

➔ Voir exercices résolus 2 et 3