

L'alternative simple et complète

L'alternative simple : La structure conditionnelle SI Alors

- Syntaxe
- SI <Condition> ALORS
 <suite d'action(s)-1>
FINSI

- La <condition> est une expression logique qui peut être vrai ou faux, selon les valeurs des paramètres la constituant.
- Si la condition est vérifiée (sa valeur est vrai), dans ce cas la <suite d'actions-1> qui sera exécutée.
- Ensuite, et à la fin de l'exécution de toutes les instructions de l'action-1, le système passe à l'exécution juste après le FINSI.

- Dans le cas contraire, c'est-à-dire lorsque la condition n'est pas vérifiée (valeur de la condition est faux).
- Le système passe directement à l'instruction qui suit le FINSI.
- La suite d'actions- 1 peut être des actions simples ou même des structures conditionnelles.

Exemple 1

- Ecrire l'algorithme qui lit un nombre réel, et affiche sur écran s'il est strictement positif.
- ALGORITHME exemple1
- VAR A : réel
- DEBUT
- ECRIRE("Donner un nombre ");
- LIRE(A);
- SI (A > 0) ALORS
- ECRIRE(A, " est positif ")
- FINSI
- FIN

L'alternative complète : La structure conditionnelle SI Alors..Sinon

- Syntaxe

- SI <Condition> ALORS
 <suite d'action(s)-1>
[SINON
 <suite d'actions(s)-2>]
FINSI

- La <condition> est une expression logique qui peut être vrai ou faux, selon les valeurs des paramètres la constituant.
- Si la condition est vérifiée (sa valeur est vrai), dans ce cas c'est la <suite d'actions-1> qui sera exécutée (et la suite d'action-2 sera ignorée).
- Ensuite, et à la fin de l'exécution de toutes les instructions de l'action-1, le système passe à l'exécution juste après le FINSI.

- Dans le cas contraire, c'est-à-dire lorsque la condition n'est pas vérifiée (valeur de la condition est faux), c'est la <suite d'actions-2> qui s'exécute, en cas où celle ci existe (facultative). Si elle n'existe pas, le système passe directement à l'instruction qui suit le FINSI.
- Les suites d'actions 1 et 2, peuvent être des actions simples ou même des structures conditionnelles.

Exemple 1

- Ecrire l'algorithme qui li un nombre réel, et dire s'il est positif ou strictement négatif.

- ALGORITHME exemple1
- VAR A : réel
- DEBUT
- ECRIRE("Donner un nombre ");
- LIRE(A);
- SI (A < 0) ALORS
- ECRIRE(A, " est négatif ")
- SINON
- ECRIRE(A, " est positif ")
- FINSI
- FIN

- *Complétez cet algorithme pour déterminer le cas où A est nulle aussi.*

- ALGORITHME POS-NEG-NUL

- VAR A : réel

- DEBUT

- ECRIRE("Donner un nombre ")

- LIRE(A) ;

- SI (A < 0) ALORS

- ECRIRE(A, " est négatif ")

- SINON {c'est-à-dire que A est ≥ 0 }

- **SI (A > 0)ALORS**

- **ECRIRE (A, " est positif ")**

- **SINON {le seule cas qui erst c'est le cas ou A = 0}**

- **ECRIRE (A, "est nulle")**

- **FINSI**

- FINSI

- FIN

Exercices

- **Exercice 1**

- 1) Ecrire l'algorithme qui permet de déterminer si un entier lu est pair ou impair.

- **Exercice 2**

- 2) Ecrire l'algorithme qui permet de déterminer si un entier lu est un multiple de 3.