

1INDF chapitre 7 : structures de contrôle

« si, sinon si, ..., sinon »

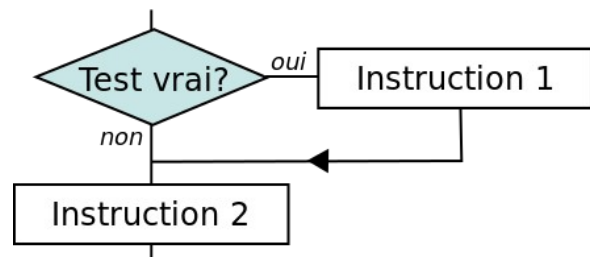
Nom:	Groupe:	Date:
------	---------	-------

Les structures de contrôle « si », « si, sinon » et « si, sinon si, sinon » sont présentées ci-dessous sous forme d'algorithme en langage naturel, de pseudo-code et de logigramme.

Test 'si' : le test si est la forme la plus simple : si *Test* est vérifié, on exécute *Instruction 1* puis *Instruction 2* ; si *Test* n'est pas vérifié, le programme exécute directement *Instruction 2*.

Pseudo code :

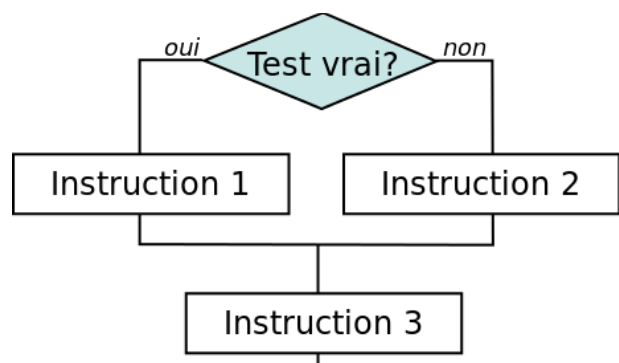
```
SI Test
    Instruction 1
FIN SI
Instruction 2
```



Test 'si, sinon' : si *Test* est vérifié, on exécute *Instruction 1* puis *Instruction 3* ; sinon on exécute *Instruction 2* puis *Instruction 3*.

Pseudo code :

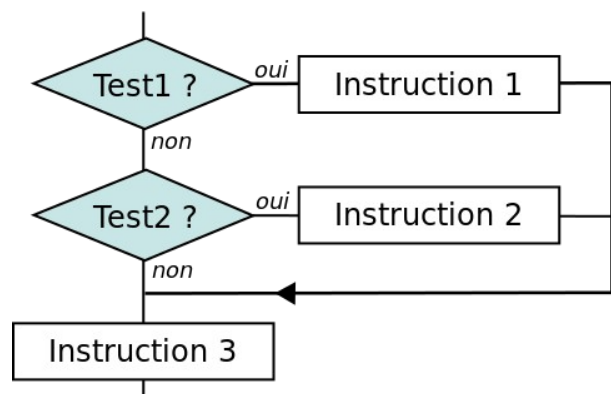
```
SI Test
    Instruction 1
SINON
    Instruction 2
FIN SI
Instruction 3
```



Test 'si, sinon si' : si *Test 1* est vérifié, le programme exécute *Instruction 1* puis *Instruction 3* ; sinon, si *Test 2* est vérifié, le programme exécute *Instruction 2* puis *Instruction 3* ; sinon le programme exécute directement *Instruction 3*.

Pseudo code :

```
SI Test 1
    Instruction 1
SINONSI Test 2
    Instruction 2
FIN SI
Instruction 3
```



A retenir

Une structure de contrôle est une instruction particulière d'un langage de programmation pouvant changer l'ordre d'exécution des instructions d'un programme et/ou définir à quelles conditions certaines instructions sont exécutées (ou pas).

En première année du collège, nous utilisons les structures de contrôle suivantes :

- les boucles 'for' et 'while', que nous verrons plus tard,
- les tests 'si', 'si, sinon' et 'si, sinon si'.

Exercice : élèves automates

Programme 1

1. initialisation : je suis assis(e), les mains posées sur les genoux.
2. s'il y a un élève à ma droite, je lève la main droite.
3. s'il y a un élève derrière moi, je mets ma main gauche sur ma tête.
4. Je me lève.

Programme 2

1. initialisation : je suis assis(e), les mains posées sur les genoux.
2. s'il y a un élève à ma droite, je lève la main droite.
3. Sinon, je mets ma main gauche sur ma tête.
4. Je me lève.

Programme 3

1. initialisation : je suis assis(e), les mains posées sur les genoux.
2. s'il y a un élève à ma droite, je lève la main droite.
3. Sinon, s'il y a un élève derrière moi, je mets ma main gauche sur ma tête.
4. Je me lève.

Questions:

- a) Quelles sont les actions et les tests exécutés dans ces programmes ?
- b) À quel type de structure de contrôle chaque programme correspond-il ?
- c) Indiquez les résultats des trois programmes faits en introduction pour la configuration de classe suivante : mettez un « vu » à droite du 😊 si l'élève lève la main droite, et un chapeau sur le 😊 si l'élève met la main gauche sur la tête.

Programme 1

😊	😊	😊	
😊		😊	😊
😊	😊	😊	😊
	😊		😊

Programme 2

😊	😊	😊	
😊		😊	😊
😊	😊	😊	😊
	😊		😊

Programme 3

😊	😊	😊	
😊		😊	😊
😊	😊	😊	😊
	😊		😊

- d) Sur une feuille à part, faites les logigrammes des trois programmes.

1INDF chapitre 7, TP3: si, sinon si, sinon

Nom:	Groupe:	Date:
------	---------	-------

Exercice 1 : la devinette

a) A votre avis, que fait le programme suivant ?

```
1.     nb_secret = 8
2.
3.     print("Devine mon nombre secret !")
4.     print("Entre un nombre au clavier:")
5.     reponse = input()
6.     reponse = int(reponse)
7.
8.     if reponse == nb_secret:
9.         print("Bravo ! Tu as deviné !")
10.    else:
11.        print("Dommage, raté...")
```

b) Quel sera l’affichage du programme si l’utilisateur entre le nombre « 4 » ? « 9 » ? « 8 » ?

c) Copiez le programme « TP3_devinette.py » depuis le répertoire >cours> sur le bureau de l’ordinateur, puis exécutez-le pour vérifier vos réponses.

d) Modifiez le programme afin de modifier le nombre secret.

e) Modifiez le programme et/ou ajoutez des lignes de commande au programme pour effectuer les tâches suivantes :

- définir un nombre secret situé entre 1 et 99 ;
- afficher à l’écran « trop grand ! » si le nombre que l’utilisateur a entré au clavier est plus grand que le nombre secret, et « trop petit ! » si le nombre est plus petit.

A retenir

En Python, l’instruction définissant un test doit se terminer par un le symbole « : », et les instructions dont l’exécution dépend d’un test sont indentées de 4 espaces, formant ainsi un **bloc** d’instructions.

Les entrées au clavier sont de type « chaîne de caractères » (string, `str`). Pour les utiliser dans des calculs, il faut les convertir en type « nombre entier » (integer, `int`) ou « nombre à virgule » (float, `float`). Le type « nombre entier » est beaucoup plus précis que le type « nombre à virgule », il est donc recommandé d’utiliser le type « nombre entier » quand c’est possible.

Exercice 2 : jeu de dé

Ecrivez un programme permettant à deux joueurs de jouer aux dés : l’ordinateur demande à chacun des joueurs d’entrer le numéro de dé qu’il a roulé, il compare les résultats puis affiche qui a gagné.

Exercice 3 : la Rousseaulette

Voici la Rousseaulette, un jeu particulièrement enlevant :

- 1. on tire au hasard un nombre entre 1 et 100 (inclus)
- 2. si le nombre est pair, on gagne 128 R\$
- 3. si le nombre est plus petit que 50, on gagne 128 R\$
- 4. sinon, on a perdu.

Programme 1

a) Combien de R\$ gagnez-vous si les nombres suivants sont tirés ?

Numéro	13	25	30	52	71	95
Gain (R\$)						

b) Quel est le numéro le plus avantageux ? Expliquez pourquoi.

.....

c) Vérifiez vos réponses en faisant tourner le programme « TP3_rousseaulette.py ».

Programme 2

Modifiez le programme ci-dessous en ajoutant le mot « sinon » pour éviter qu'un joueur ne rapporte deux gains avec le même numéro :

- 1. on tire au hasard un nombre entre 1 et 100 (inclus)
- 2. si le nombre est pair, on gagne 128 R\$
- 3. si le nombre est plus petit que 50, on gagne 128 R\$
- 4. sinon, on a perdu.

a) Quels seront les gains avec la nouvelle condition si les mêmes numéros sont tirés ?

Numéro	13	25	30	52	71	95
Gain (R\$)						

b) Modifiez le programme « TP3_rousseaulette.py » pour inclure la nouvelle condition. Vous devez créer une copie du fichier et sauvegarder la nouvelle version sous le nom « TP3_rousseaulette_v2_NDF.py ».

Programme 3

a) Remplacez les étapes 2 et 3 du programme par une seule étape contenant l'opérateur « ou »

- 1. on tire au hasard un nombre entre 1 et 100 (inclus)
- 2.
- 3. sinon, on a perdu.

b) Dans le programme de la question (a), remplacez le « ou » par un « et » :

1. on tire au hasard un nombre entre 1 et 100 (inclus)

2.

3. sinon, on a perdu.

c) Avec le programme (b), quels seront les gains si les mêmes numéros sont tirés ?

Numéro	13	25	30	52	71	95
Gain (R\$)						

d) Modifiez le programme « *TP3_rousseaulette.py* » pour faire les versions « *TP3_rousseaulette_OU_NDF.py* » et « *TP3_rousseaulette_ET_NDF.py* ».

Exercice 5 : note arrondie à la demi-bonne

Le tableau ci-dessous montre des exemples de notes finales telles que calculées au TP1.

a) Complétez la deuxième ligne du tableau avec les notes arrondies à la demi-bonne.

Note finale	3,18	3,78	4,74	5,23	5,25	5,75
Note arrondie						

b) Décrivez en quelques mots la méthode que vous avez utilisée pour arrondir les notes.

.....

c) Copiez le programme « *TP4_notearrondie.py* » sur le bureau de l'ordinateur, et exécutez-le en entrant au clavier chacune des notes utilisées à l'exercice (a).

d) Le programme est-il juste ? *Justifiez.*

.....

e) Modifiez le programme pour que son résultat soit toujours juste. *Sauvegardez votre programme sous le nom « *TP4_notearrondie_qe_NDF.py* ».*

f) Modifiez le programme afin d'utiliser la structure « if, elif, elif, ..., else ». *Sauvegardez votre programme sous le nom « *TP4_notearrondie_qf_NDF.py* ».*