

## Exercices : Algorithmique et programmation :

---

**Test :** Pour chaque candidat au bac, en fonction de sa moyenne , un programme affiche "admis" **si la moyenne est supérieure ou égale à 10**, "Participe aux oraux de rattrapage" **si la moyenne est comprise entre 8 et 10 exclu**, "Refusé" **si la moyenne est strictement inférieure à 8**

En pseudo-code cela donne :

```
moyenne <- Entrer("Donnez la moyenne du candidat")
Si moyenne >= 10
    afficher("Admis")
Fin De Si
Si moyenne < 10 et moyenne >= 8
    afficher("Rattrapage")
Fin De Si
Si moyenne < 8
    afficher("Refusé")
Fin De Si
```

### Commentaires :

1.  $moyenne \geq 10$  est une **expression** qui prend une des deux valeurs soit Vrai soit Faux, en fonction du contenu de la variable moyenne au moment où cette expression est évaluée
2. Si c'est vrai alors le bloc d'instructions (ici il n'y a qu'une instruction afficher("Admis")) délimité entre "Si moyenne  $\geq 10$ " et "Fin de Si" est exécuté
3. Si c'est faux l'instruction suivante est Si moyenne  $< 10$  et moyenne  $\geq 8$
4.  $moyenne < 10$  et moyenne  $\geq 8$  est vrai que si les deux expressions moyenne  $< 10$  et moyenne  $\geq 8$  le sont

Voici la traduction en Python du premier algorithme :

```
moyenne = float(input("Donnez la moyenne du candidat"))
if moyenne >= 10:
    print("Admis")
if moyenne >= 8 and moyenne < 10:
    print("Rattrapage")
if moyenne < 8 :
    print("Refusé")
```

### Commentaires :

1. une moyenne peut être un nombre à virgule aussi la fonction `float()` permet de convertir ce qui est entré au clavier en nombre à virgule
2. Il n'y a pas de Fin de Si en Python, **l'indentation**, le fait que les lignes sont décalées vers la droite d'un multiple de 4 caractères, permet de reconnaître le bloc d'instructions qui doit être exécuté après le symbole :
3. On aurait pu écrire à la place de l'expression moyenne  $\geq 8$  and moyenne  $< 10$ , comme en mathématiques l'expression  $8 \leq \text{moyenne} < 10$

Dans cet algorithme si une moyenne est supérieure ou égale à 10 les expressions moyenne  $< 10$  et moyenne  $\geq 8$  puis moyenne  $< 8$  seront quand même évaluées, on peut éviter cela avec la version suivante :

```
moyenne <- Entrer("Donnez la moyenne du candidat")
Si moyenne >= 10
    afficher("Admis")
Sinon
    Si moyenne < 10 et moyenne >= 8
        afficher("Rattrapage")
    Sinon
        afficher("Refusé")
    Fin De Si
Fin de Si
```

### Commentaires :

1. Cette fois ci si la moyenne est supérieure ou égale à 10 le bloc d'instructions entre Sinon et le dernier Fin de Si n'est pas exécuté
2. **Gagne-t-on vraiment du temps** à faire cela? Est ce aussi **lisible** que le précédent algorithme? A suivre...

En Python cela donne :

```
moyenne = float(input("Donnez la moyenne du candidat"))
if moyenne >= 10:
    print("Admis")
else:
    if 8 <= moyenne < 10:
        print("Rattrapage")
    else :
        print("Refusé")
```

ou encore de manière plus concise :

```

moyenne = float(input("Donnez la moyenne du candidat"))
if moyenne >= 10:
    print("Admis")
elif 8 <= moyenne < 10:
    print("Rattrapage")
else :
    print("Refusé")

```

### Commentaires :

1. elif est une contraction de else if c'est à dire sinon si. Ceci permet de tester plusieurs possibilités , else ou sinon correspond au dernier cas

**Ex n°1** : Faire évoluer le dernier programme Python en rajoutant les possibilités suivantes en utilisant elif :

Si la moyenne  $\geq 16$  alors afficher("Mention Très Bien"), Si la moyenne est comprise entre 14 et 16 afficher("Mention Bien"), Si la moyenne est comprise entre 12 et 14 afficher("Mention Assez Bien"), Si la moyenne est comprise entre 10 et 12 afficher("Mention Passable")

**Attention ! en pseudo-code  $\leftarrow$  est le symbole d'affectation et  $=$  le symbole d'égalité, alors qu'en Python  $=$  est le symbole d'affectation et  $==$  le symbole d'égalité**

Voir ci-dessous

```

mot_de_passe <- "Python"
essai <- Entrer(" Quel est le mot de passe ?")
Si essai = mot_de_Passe
    afficher("Bienvenue")
Sinon
    afficher("Accès Refusé")
Fin de Si

```

en Python :

```

mot_de_passe = "Python"
essai = Entrer(" Quel est le mot de passe ?")
if essai == mot_de_Passe:
    print("Bienvenue")
else:
    print("Accès Refusé")

```

On vient de voir ci-dessus qu'à la variable mot\_de\_passe est associé un nouveau type de données appelé chaîne de caractères, str pour string en Python

**Ex n°2 :** Dans l'interpréteur Python exécuter  $4\%2$  puis  $6\%2$  puis  $3\%2$  puis  $5\%2$ . A quoi peut nous aider l'opération  $\%$  ?

**Ex n°3 :** Ecrire un algorithme puis un programme qui prend en entrée un nombre entier si ce nombre est pair vous affichez le résultat de la division par 2 de ce nombre sinon vous affichez le résultat de la soustraction de 1 à ce nombre

**Ex n°4 :** Ecrire un algorithme puis un programme qui prend en entrée deux nombres et affiche le plus grand des deux

**Ex n°5 :** Ecrire un algorithme puis un programme qui prend en entrée un nombre et affiche ce nombre s'il est positif , son opposé s'il est strictement négatif

**Ex n°6 :** Que fait l'algorithme suivant ?

```
a <- Entrer(" Donnez un  nombre entier ")
Si a est pair
    a <- a / 2
Sinon
    a <- 3*a + 1
```

Si un nombre est impair , lorsqu'on le multiplie par 3 et si on rajoute 1 obtient-on forcément un nombre pair? (indication : un nombre entier  $x$  est impair s'il existe n **entier** tel que  $x = 2n + 1$  , un nombre entier  $x$  est pair s'il existe n **entier** tel que  $x = 2n$

**Ex n°7 :** Etant donné le quadrilatère ABCD tel que les coordonnées de A,B,C et D sont  $(x_A; y_A)$ ,  $(x_B; y_B)$ ,  $(x_C; y_C)$  et  $(x_D; y_D)$  donner un algorithme puis un programme qui affiche si ABCD est un parallélogramme ou pas