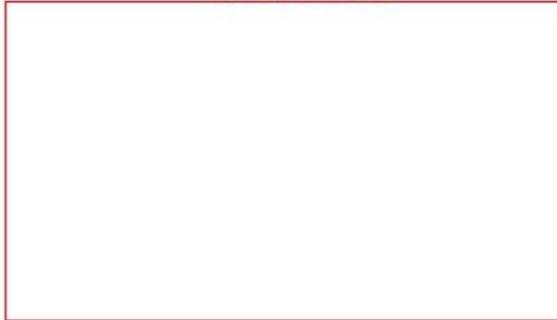


Python

Mémoire



M. Micro-processeur

1) Algorithme :

a) Affectation :

Syntaxe des instructions utiles dans cette fiche :

Langage naturel	Python
Affecter à A la valeur 5	A=5
Saisir x	<code>x = input('message...')</code> ↑
Afficher A	<code>print(A)</code>
Quotient de la division euclidienne de A par B	A//B
Reste de la division euclidienne de A par B	A%B
Racine carrée de A	<code>sqrt(A)</code> <i>Dans la console, on commencera par importer la fonction sqrt</i> from math import sqrt
Reste de la division euclidienne de A par B	A%B

Exercice d'application :

Un commerçant accorde une remise sur des articles. On souhaite connaître le montant de la remise en euros.

Voici un algorithme donnant la solution au problème :

Saisir le prix de départ A
Saisir le pourcentage de remise P

Affecter au montant de la remise R la valeur $A \times \frac{P}{100}$

Afficher R

- 1) a) Calculer la valeur de la variable R lorsque A = 56 et P = 30.
b) Donner une interprétation concrète du résultat précédent.
- 2) Même question avec A = 13 et P = 45.
- 3) Compléter l'algorithme pour afficher également le prix à payer B.
- 4) a) Calculer la valeur des variables R et B lorsque A = 159 et P = 24.
b) Donner une interprétation concrète des résultats précédents.

b) Conditions :

Syntaxe :

Langage naturel	Python
Si <i>Condition</i> Alors <i>Instructions1</i> Sinon <i>Instructions2</i> Fin Si	if <i>condition:</i> <i>Instruction1</i> else: <i>Instruction2</i>
Quotient de la division euclidienne de A par B	A/B
Reste de la division euclidienne de A par B	A%B

Exercice d'application :

Écrire et tester un programme qui demande en entrée à un client le montant total de ses achats.

En fonction de la somme dépensée, le programme affiche en sortie le prix à payer :

- Si la somme dépensée est strictement inférieure à 75 €, il obtient 5 % de remise.
- Si la somme dépensée est supérieure à 75 €, il obtient 8 % de remise.

c) Boucles :

Syntaxe :

Langage naturel	Python
Tant que <i>Condition est vraie</i> <i>Instructions</i> Fin Tant que	while <i>Condition:</i> <i>Instructions</i>
Pour <i>i allant de 3 à 7</i> <i>Instructions</i> Fin Pour	for <i>i in range(3,8):</i> <i>Instructions</i>

En Python, **range(3,8)** désigne la séquence des entiers n vérifiant $3 \leq n < 8$.
range(5) désigne la séquence des entiers 0, 1, ..., 4.

Sortir de la boucle	break
Générer un nombre entier aléatoire de l'intervalle [1;5]	randint(1,5)

Exercice d'application :

On dépose 25€ dans une tirelire.

L'algorithme suivant, écrit en langage naturel, permet de calculer le nombre de pièces de 1€ ou 2€ ajoutés de façon aléatoire dans la tirelire avant de dépasser 50€.

- 1) Compléter l'algorithme.
- 2) Pourquoi le programme affiche en sortie « D-1 » ?
- 3) Ecrire et tester un programme traduisant cet algorithme.

```

Affecter à S la valeur 25
Affecter à D la valeur 0
Tant que S<...
    Affecter à A la valeur aléatoire 1 ou 2
    Affecter à S la valeur ...
    Affecter à D la valeur D+1
    Afficher A
Fin Tant que
Afficher D-1
    
```

2) De l'algorithme à Python:

On considère la suite (u_n) définie par $u_1 = 4$ et $u_{n+1} = 7u_n + 5$, pour tout entier naturel n .

Écrire un algorithme permettant de calculer le terme u_n où n est un entier supérieur ou égal à 2.

i L'utilisateur rentre la valeur de n et l'ordinateur affiche u_n .

Initialisation	?	Pour k allant de 2 à n
Traitement	?	fin Pour
	?	$u \leftarrow 4$
	?	$u \leftarrow 7u + 5$

1) Compléter l'algorithme ci-dessous

2) Implémenter

3)

On considère la suite (u_n) définie par $u_1 = 4$ et $u_{n+1} = 7u_n + 5$, pour tout entier naturel n .

Écrire un algorithme permettant de calculer le terme u_n où n est un entier supérieur ou égal à 2.

i L'utilisateur rentre la valeur de n et l'ordinateur affiche u_n .

Initialisation	fin Pour	[1]
Traitement	Pour k allant de 2 à n	[2]
	$u \leftarrow 4$	[3]
	$u \leftarrow 7u + 5$	[4]

Implémenter l'algorithme en langage Python.