

Algorithmique 1 : Notions de base

Tronc commun mathématique et informatique



Dr BENMAHDI-HABRI Meryem Bochra

Table des matières



I - Chapitre 1 : Le monde des variables	3
1. Introduction	3
2. Notion de variable	3
3. Nommage d'une variable	4
3.1. Ce qu'il faut savoir sur le nommage des variables	4
3.2. Exercice : Application des règles :	4
4. Les types de variables	4
5. Déclaration d'une variable	5
6. Affectation	5
7. Initialisation	6
8. Affichage	7
8.1. L'instruction de l'affichage	7
8.2. Exemple pour l'affichage d'une seule variable :	7
8.3. Afficher plusieurs variables avec un seul printf	7
9. La lecture des variables :	8
10. Exemples pour conclure :	8
11. Quiz pour valider le chapitre 2	9
Webographie	11

Chapitre 1 : Le monde des variables



1. Introduction

Pour effectuer un stockage de valeurs, un calcul ou un traitement, il faut un enregistrement temporaire de données. Cet enregistrement nécessite la déclaration d'un endroit dans la mémoire qui servira au stockage de la variable.

2. Notion de variable

Une variable est une information temporaire qu'on stocke dans la mémoire c'est à dire sa valeur peut changer pendant le déroulement du programme d'où son appellation.

Dans langage C, une variable est constituée de deux choses :

1. un nom : c'est ce qui permet de reconnaître une variable.
2. une valeur : c'est le nombre qu'elle stocke, par exemple 13 ;

3. Nommage d'une variable

3.1. Ce qu'il faut savoir sur le nommage des variables

Dans la plus part des langage de programmation, il y a quelques règles, contraintes et conventions à respecter lors du nommage des variables :

- L'implication que des chiffres est impossible.
- Un nom de variable doit commencer par une lettre ;
- Un nom de variable peut contenir que des minuscules ou que des majuscules et il peut aussi contenir la majuscules, les minuscule et les chiffres en même temps (abcABC012) ;
- L'implication que des chiffres est impossible;
- Les espaces sont interdits. À la place, on peut utiliser le caractère "underscore " _ (tiret de huit) C'est le seul caractère différent des lettres et chiffres autorisé ;
- L'utilisation des accents (èâê etc.) n'est pas possible.

3.2. Exercice : Application des règles :

Cocher les nom de variables valides:

- noteInfo
- mail
- nom
- Bac2020
- numero_de_telephone
- Prix-TTC
- prénom
- main

4. Les types de variables

A la création d'une variable, Il est nécessaire d'indiquer son type. Il existe plusieurs types de variables.

Voici les principaux types de variables existant en langage C :

Type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	3.4×10^{-38}	3.4×10^{38}
double	1.7×10^{-308}	3.4×10^{38}

Remarque

Les types les plus utilisés dans la suite de ce cours sont: int pour les nombres entiers et double pour les nombres réels.

5. Déclaration d'une variable

Pour utiliser une variable dans un programme, on doit d'abord la déclarer. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

Exemple

```
1 int main ()
2 {
3 //Type nom_de_la_variable
4 int quantite ; // déclaration de la variable quantite de type entier
5 double prix ;
6 int somme , produit ;
7 return 0;
8 }
```

6. Affectation

Affecter une valeur à une variable c'est à dire lui donner une valeur. Pour effectuer une affectation il suffit d'indiquer simplement le nom de la variable, puis un signe égal (=), et enfin la valeur qu'on veut y mettre.

Exemple

```
1 # include <stdio .h>
2 # include <stdlib .h>
3 int main ()
4 {
5 int age ;
6 age = 21;
7 return 0;
```

```
8 }
```

7. Initialisation

- Initialiser une variable c'est à dire lui donner une valeur initiale. Au moment de la déclaration d'une variable, quelle valeur a-t-elle?
- Initialiser une variable ça consiste à combiner la déclaration et l'affectation de cette variable dans la même instruction.
- Si on n'affecte pas une valeur initiale à une variable, elle peut prendre n'importe quelle valeur.
- L'avantage, c'est qu'on est sûrs que cette variable contient une valeur correcte, et pas du n'importe quoi.

Exemple

```
1 # include <stdio .h>
2 # include <stdlib .h>
3 int main ()
4 {
5 int AGE = 24;
6 AGE = 26
7 AGE = 25;
8 AGE = 28;
9 return 0;
10 }
```

8. Affichage

8.1. L'instruction de l'affichage

L'instruction qui sert à l'affichage est l'instruction printf.

Avec cette instruction, on peut afficher un simple message et on peut aussi afficher une variable ou plusieurs variables. Mais dans l'affichage d'une variable on doit ajouter un symbole spécial '%' à l'endroit où l'on veut afficher la valeur de la variable.

- Format pour chaque type:

Format	Type attendu
"%d"	int
"%ld"	long
"%f"	float
"%f"	double
"%c"	char

8.2. Exemple pour l'affichage d'une seule variable :

```

1 int main ()
2 {
3 int age = 24;
4 printf ("la variable age contient %d comme valeur .", age);
5 return 0;
6 }

```

- Ce programme affiche: la variable age contient 24 comme valeur .
- La lettre après le % permet d'indiquer ce que l'on doit afficher. 'd' signifie que l'on souhaite afficher un int.

8.3. Afficher plusieurs variable avec un seul printf

Il est possible d'afficher la valeur de plusieurs variables dans un seul printf. Il suffit pour cela d'indiquer des % c ou des % c là où vous voulez, puis d'indiquer les variables correspondantes dans le même ordre, séparées par des virgules.

Exemple

```

1 int main ()
2 {
3 int nombreEtudiants = 240;
4 double moyenneBac = 14.5 ;
5 printf ("Il y a %d etudiants inscrits avec une moyenne de %f en Bac",
6         nombreEtudiants , moyenneBac );
6 return 0;
7 }

```

Ce programme affiche:

Il y a 240 étudiants inscrits avec une moyenne de 14.5 en Bac

9. La lecture des variables :

- La lecture d'une variable signifie la saisie de la valeur de la variable via le clavier. L'instruction qui permet cela c'est l'instruction : scanf.
- la syntaxe de cette fonction ressemble beaucoup à printf. On doit mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float,...)
- Puis on doit indiquer le nom de la variable qui va recevoir cette valeur.

Exemple

```
1 int main ()
2 {
3 int age = 0;
4 scanf ("%d", &age);
5 return 0;
6 }
7
```

- On doit mettre le %d entre guillemets.
Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.

10. Exemples pour conclure :

Les exemples suivants résument ce qu'on va voir dans ce chapitre

Exemple

Un petit programme simple qui demande l'âge de l'utilisateur et qui le lui affiche ensuite :

```
1 int main ()
2 {
3 int age = 0; // On initialise la variable a 0
4 printf (" Quel age avez - vous ? ");
5 scanf ("%d", &age); // On demande d'entrer l'age avec scanf
6 printf ("Ah ! Vous avez donc %d ans !\n\n", age);
7 return
```

Exemple

Ce programme permet de lire deux entiers et d'afficher leur produit.

```
1 int main ()
2 {
3 int resultat = 0, nombre1 = 0, nombre2 = 0;
4 // On demande les nombres 1 et 2 a l'utilisateur :
5 printf (" Entrez le nombre 1 : ");
6 scanf ("%d", & nombre1 );
7 printf (" Entrez le nombre 2 : ");
8 scanf ("%d", & nombre2 );
```



```
9 // On fait le calcul :
10 resultat = nombre1 * nombre2 ;
11 // Et on affiche le produit a l'ecran :
12 printf ("%d ^* %d = %d\n", nombre1 , nombre2 , resultat );
13 return 0;
14 }
```

11. Quiz pour valider le chapitre 2

Exercice

Est ce qu'on peut nommer une variable avec un nom scanf?

- Non
- Oui

Exercice

Est ce qu'on peut nommer une variable avec un nom 2_age?

- Non
- Oui

Exercice

On déclare la variable moyenne du bac avec:

- int moyennedubac;
- double moyennedu bac;
- double moyenne_du_bac;

Exercice

Qu'affiche le programme suivant:

```
# include <stdio .h>
# include <stdlib .h>

int main ()
{
int nombreEtudiants = 240;

printf ("Il y a %d etudiants inscrits .", nombreEtudiants );

return 0;

}
```

- Il y a 240 etudiants inscrits

Webographie



[Cours Algorithmique Messabihi première année MI]

<https://sites.google.com/site/informatiquemessabihi/>

[Le langage C en 20h]

<https://c.developpez.com/cours/20-heures/>

[Open class room]

<https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/13300-vous-avez-dit-programmer>

