



**RESSOURCE R108-INFO1**  
ALGORITHMIQUE et PROGRAMMATION en Langage C



5 TD et 10.5 TP

Année 2022/2023

**Dominique Garric**  
[garric@u-pec.fr](mailto:garric@u-pec.fr)

**& Gabriel Soranzo**  
[gabriel.soranzo@u-pec.fr](mailto:gabriel.soranzo@u-pec.fr)

**& José Martin**  
[jose.martin@u-pec.fr](mailto:jose.martin@u-pec.fr)

---

---

---

---

---

---

---

---

MEMORISER

COMMUNIQUER


→ CALCULER

INSTRUCTIONS niveau débutant

1- MEMORISER

2- COMMUNIQUER

3- CALCULER



---

---

---

---

---

---

---

---

MEMORISER


COMMUNIQUER

→ CALCULER

INSTRUCTIONS niveau débutant

1- MEMORISER

1.1- Caractérisation des données



---

---

---

---

---

---

---


---

Les DONNEES pour MEMORISER

DONNEES

Variables

Constantes



MEMORISER → COMMUNIQUER → CALCULER

4

---

---

---


---

---

---


---

MEMORISER : DEFINITION d'une VARIABLE



MEMORISER → COMMUNIQUER → CALCULER

5

 compt

Une variable permet de manipuler des quantités de diverses natures (valeurs entières, réelles, caractères...) dans un programme.

Une **variable** est une donnée dont la valeur peut varier en cours d'exécution du programme : sa valeur peut être changée plusieurs fois par les instructions d'un programme.

---

---

---


---

---

---

---

MEMORISER : DEFINITION d'une CONSTANTE



MEMORISER → COMMUNIQUER → CALCULER

6

&PI → 

3.14

 PI

Une constante permet de manipuler des quantités de diverses natures (valeurs entières, réelles, caractères...) dans un programme.

Une **constante** est une donnée dont la valeur ne peut pas varier en cours d'exécution du programme : sa valeur est fixée en début de programme et elle ne peut plus être changée par les instructions.

---

---

---

---

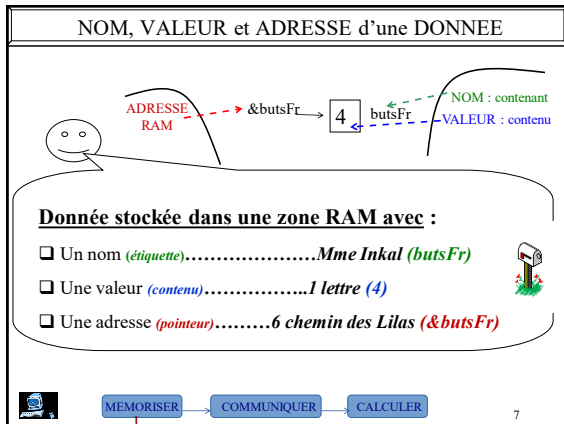
---

---

---

Dominique Garric – IUT Sénart-Fontainebleau – Département GEII

2



---

---

---

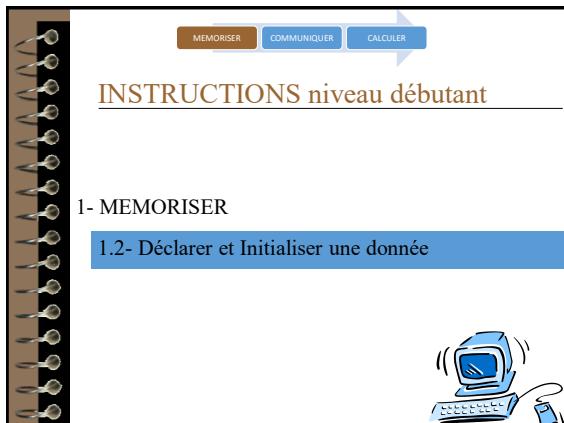
---

---

---

---

---



---

---

---

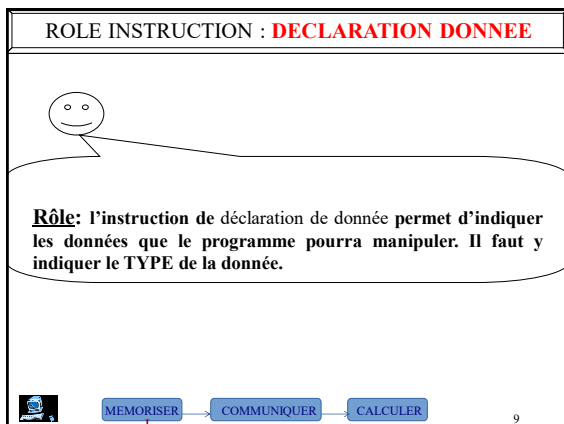
---

---

---

---

---



---

---

---

---

---

---

---

---

MEMORISER : les 2 TYPES de DONNEES numériques

```
graph TD; ENTIER --> binaire; ENTIER --> REEL;
```

10

---

---

---

---

---

---

---

MEMORISER : TYPES de DONNEES

Le **Type** d'une donnée définit *sa nature* (les valeurs qu'elle peut prendre et les opérations qu'elle peut subir) :

**ATTENTION** aux dépassements de capacité !!

La taille de sa zone mémoire ..... 2 octets

L'ensemble des valeurs possibles ..... [0,65535]

Les opérateurs applicables à cette donnée ..... + - / \* %

11

---

---

---

---

---

---

---

MEMORISER : Types ENTIER

```
graph TD; ENTIER --> N1[entiers naturels]; ENTIER --> N2[entiers relatifs]; N1 --> N1a["[0,+∞] Nombres non signés"]; N1a --> N1a2[2 octets]; N1a --> N1a4[4 octets]; N1a2 --> N1a2a["[0,65535] Entier non signé court"]; N1a4 --> N1a4a["[0,4 294 967 295] Entier non signé long"]; N2 --> N2a["[-∞,+∞] Nombres signés"]; N2a --> N2a2[2 octets]; N2a --> N2a4[4 octets]; N2a2 --> N2a2a["[-32768,32767] Entier signé court"]; N2a4 --> N2a4a["[-2 147 483 647] Entier signé long"];
```

12

---

---

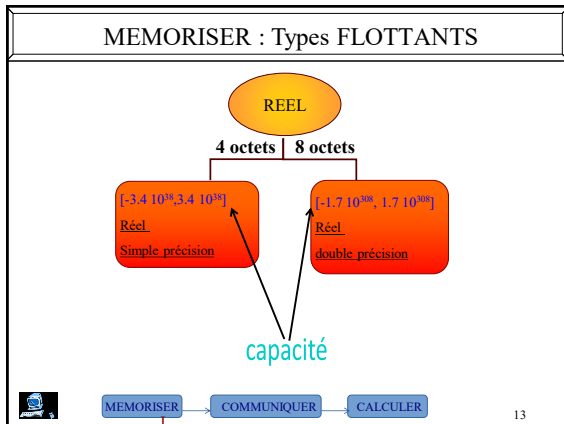
---

---

---

---

---




---

---

---

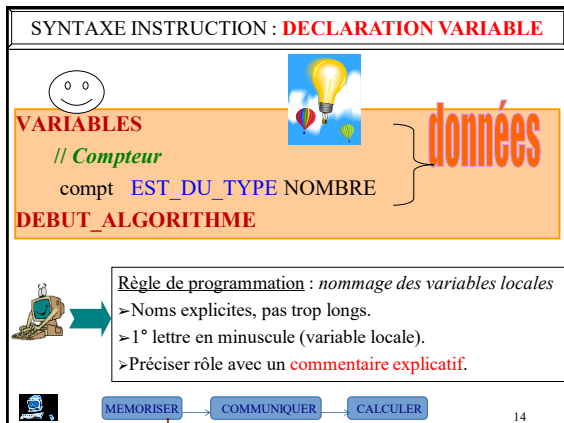
---

---

---

---

---




---

---

---

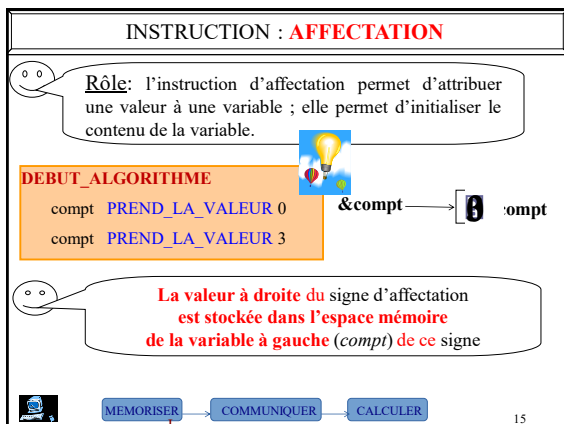
---

---

---

---

---




---

---

---

---

---



---

---


---

SYNTAXE INSTRUCTION : **DECLARATION CONSTANTE**


(Instruction utilisable en langage C)



**const**    *NOMCONSTANTE*=valeur : type  
**const**    PI= 3.14 : REEL // nombre PI pour calcul aire disque



Règle de programmation : *nommage des constantes*  
➤ Noms explicites, pas trop longs  
➤ Tout en MAJUSCULES  
➤ Préciser rôle avec un **commentaire explicatif**



MEMORISER → COMMUNIQUER → CALCULER

16

---

---

---

---

---

---

---

---

Démo Permutation

Le programme initialise 2 variables de type entier par affectation, puis il échange la valeur des 2 variables



Comment ne pas écraser/oublier des valeurs mémorisées ??

---

---

---

---

---

---

---

---

APPLICATION

PERMUTATION	ALGO à FAIRE
Permuter les valeurs de a et b (quels que soient leurs valeurs)	➤ Lister les variables utiles ➤ Trouver les étapes pour échanger les valeurs des 2 variables

---

---

---

---

---

---

---

---


MEMORISER

COMMUNIQUER

CALCULER

### INSTRUCTIONS niveau débutant

2- COMMUNIQUER



---

---

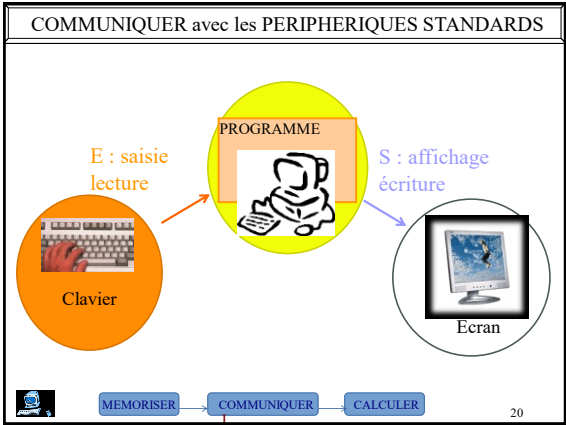
---

---

---

---

---



---

---

---

---

---

---

---

## Démo Communication Clavier / Ecran

Le programme demande à l'utilisateur le score d'un match de football, puis il réaffiche ce score sur l'écran



La communication Clavier / Ecran permet au programme de dialoguer avec l'utilisateur

---

---

---

---

---

---

---

MEMORISER


COMMUNIQUER

CALCULER

### INSTRUCTIONS niveau débutant

#### 2- COMMUNIQUER

2.1- Ecrire sur l'écran de l'ordinateur



---

---

---

---

---

---

---

ROLE INSTRUCTION : AFFICHAGE ECRAN

PROGRAMME

S :  
affichage  
écriture

Ecran

L'affichage écran permet d'écrire du texte sur l'écran pour :

- Informer l'utilisateur,
- Lui demander des informations.

Nom: \_

Ecran

MEMORISER

COMMUNIQUER

CALCULER

23

---

---

---

---

---

---

---

SYNTAXE INSTRUCTION : AFFICHAGE ECRAN

Donner texte ou nom variable à afficher :

VARIABLES

// Age des étudiants

age EST\_DU\_TYPE NOMBRE

DEBUT\_ALGORITHME

age PREND\_LA\_VALEUR 18

// affichage texte seul

AFFICHER " \nVotre age est : " //\n : passage à la ligne

// affichage valeur variable

AFFICHER age

FIN\_ALGORITHME

MEMORISER

COMMUNIQUER

CALCULER

24

---

---

---

---

---

---

---



MEMORISER


COMMUNIQUER

CALCULER

### INSTRUCTIONS niveau débutant

2- COMMUNIQUER

2.2- Lire une information sur le clavier



---

---

---

---



---

---

---

ROLE INSTRUCTION : SAISIE CLAVIER

E : lecture



La saisie clavier permet de lire des valeurs tapées au clavier par l'utilisateur du programme, afin de les affecter à une variable.

La saisie clavier permet d'initialiser des variables.

3 age

MEMORISER

COMMUNIQUER

CALCULER

26

---

---

---


---

---

---

---

SYNTAXE INSTRUCTION : SAISIE CLAVIER



Donner variable qui reçoit la saisie :

VARIABLES

age EST\_DU\_TYPE NOMBRE // Age des étudiants

DEBUT\_ALGORITHME

// demande à l'utilisateur du programme

AFFICHER " Quel est votre age : "

// récupération de la valeur tapée au clavier par l'utilisateur

LIRE age // !! ALGOBOX affiche automatiquement le message de demande de valeurs, pas le C !!

FIN\_ALGORITHME

MEMORISER

COMMUNIQUER

CALCULER

27

---

---

---

---

---

---

---

# APPLICATION

MONDIAL 2018	ALGO à FAIRE
le programme saisit et réaffiche le score de la finale France-Croatie	<ul style="list-style-type: none"><li>➤ Déclaration des variables pour stocker les buts des 2 camps</li><li>➤ Saisies au clavier du nombre de buts marqués par la France, puis par la Croatie</li><li>➤ Réaffichage sur l'écran du score final du match</li></ul>

---

---

---

---

---


---

---

MEMORISER COMMUNIQUER CALCULER

## INSTRUCTIONS niveau débutant

3- CALCULER



---

---

---

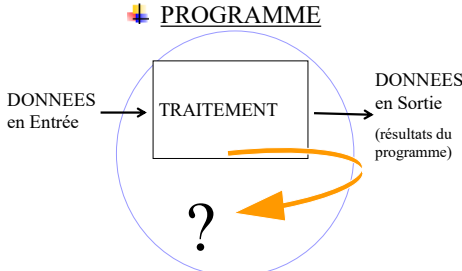
---

---

---

---

NECESSITE des INSTRUCTIONS de CALCUL



PROGRAMME

DONNEES en Entrée → TRAITEMENT → DONNEES en Sortie (résultats du programme)

?

MEMORISER COMMUNIQUER CALCULER

30

---

---

---

---

---

---

---

MEMORISER


COMMUNIQUER

CALCULER

### INSTRUCTIONS niveau débutant

3- CALCULER

3.1- SYNTAXE Instruction de CALCUL



---

---

---

---


---

---

---

0 0

SYNTAXE INSTRUCTION : **CALCUL**



```
VARIABLES // variables pour calcul moyenne
a EST_DU_TYPE NOMBRE
b EST_DU_TYPE NOMBRE
moy EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME
// valeur des données à moyenner
a PREND_LA_VALEUR 1
b PREND_LA_VALEUR 2
// calcul : nomVariableResultat PREND_LA_VALEUR Formule de Calcul
moy ← PREND_LA_VALEUR (a+b)/2

FIN_ALGORITHME
```

**Attention !!**  
**METTRE la VARIABLE RESULTAT à gauche !!**

MEMORISER

COMMUNIQUER

CALCULER

32

---

---

---

---

---

---

---

FORMULE de CALCUL

Une FORMULE de CALCUL

EXPRESSION de CALCUL  
avec opérateurs

FONCTION STANDARD  
mathématiques...

a+b

cos(teta)

MEMORISER

COMMUNIQUER

CALCULER

33

---

---

---

---

---

---

---

EXPRESSION ARITHMETIQUE


Expression Arithmétique

➤ Calculs en base 10

➤ Opérateurs Arithmétiques

➤ a, b, c, résultats : entiers ou réels

(axb+3) / (c-5)  
a%b



MEMORISER → COMMUNIQUER → CALCULER

34

---

---

---

---

---


---

---

---

OPERATEURS ARITHMETIQUES

Opérateur	Nom	En C
+	Addition	+
-	Soustraction	-
X	Multiplication	*
/	Division	/
%	Modulo	%



MEMORISER → COMMUNIQUER → CALCULER

35

---

---

---

---

---

---

---

---

MEMORISER


COMMUNIQUER

CALCULER

INSTRUCTIONS niveau débutant

3- CALCULER

3.2- EXECUTION CPU Instruction de CALCUL



---

---

---

---

---

---

---

---

EXECUTION INSTRUCTION : **CALCUL**

resu **PREND\_LA\_VALEUR** a-b //soustraction

2


↓

STOCKAGE (affectation) du  
RESULTAT  
dans la variable *resu*

1

↘

CALCUL (évaluation) du  
RESULTAT de  
a-b

 MEMORISER → COMMUNIQUER → CALCULER

37

---

---

---

---

---

---

---

---

ETAPES EXECUTION INSTRUCTION : **CALCUL**

a, b, resu : entier

resu **PREND\_LA\_VALEUR** a-b

1

1) Calcul du résultat : évaluation de la  
valeur de (a-b) par l'ALU :

2

2) Stockage du résultat : affectation de la  
valeur obtenue dans la variable *resu* :

a → 3

&a →

b → 2

&b →


ALU


↑

3 2

resu → 1

&resu →



 MEMORISER → COMMUNIQUER → CALCULER

38

---

---

---

---

---

---

---

---

CALCULER : REGLES d'EVALUATION

COMBIEN VAUT *resu* ?

→


$resu = (4 + 2) \times 2$

→

$resu = 4 + 2 \times 2$  **8 ou 12??**

→

$resu = 4 / 2 / 2$  **4 ou 1??**

 MEMORISER → COMMUNIQUER → CALCULER

39

---

---

---

---

---


---

---

---

PRIORITES des OPERATEURS en C

Priorité	Opérateur
1	( )
2	NON (non logique) — (complément à 1)
3	x (multiplication) / (division) % (modulo)
4	+ (addition) - (soustraction)
5	< > ≥ ≤
6	== (égalité) ≠ (différent)
7	. (et binaire)
8	⊕ (ou exclusif)
9	+ (ou binaire)
10	ET (et logique)
11	OU (ou logique)
12	= (affectation)



MEMORISER → COMMUNIQUER → CALCULER

40

---

---

---

---

---


---

---

---

CALCULER : REGLES d'EVALUATION

$$\text{resu} = \overbrace{(4 + 2)}^6 \times 2 \quad \text{12}$$
$$\text{resu} = 4 + \overbrace{2 \times 2}^4 \quad \text{8}$$
$$\text{resu} = \overbrace{4 / 2 / 2}^2 \quad \text{1}$$



MEMORISER → COMMUNIQUER → CALCULER

41

---

---

---

---


---

---

---


---

CALCULER : REGLES d'EVALUATION



L'évaluation d'un calcul par le CPU suit les règles :

1. L'ordre de calcul dépend des **parenthèses** (forcent les priorités).
2. L'ordre de calcul dépend, ensuite, de la **priorité des opérateurs**.
3. Pour un même niveau de priorité, l'évaluation de l'expression se fait **de gauche à droite** (à partir de l'opérateur d'affectation =).



MEMORISER → COMMUNIQUER → CALCULER

42

---

---

---

---


---

---

---

---

# APPLICATION



<b>RESISTANCE //</b>	<b>ALGO à FAIRE</b>
<p>Le programme calcule la résistance équivalente à trois résistances montées en parallèle</p>	<p>➤ trois valeurs de résistances sont initialisées en milliOhms</p> <p>➤ La résistance équivalente est calculée en Ohms, à partir des valeurs précédentes (attention aux unités!!). Pour un montage parallèle :</p> <p><math>1/req = 1/r1 + 1/r2 + 1/r3</math></p>

---

---

---

---

---

---

---