

Chapitre 2

Les instructions



- 1) L'instruction d'affectation
- 2) Les instructions d'entrée/sortie
- 3) Les structures conditionnelles
- 4) Les structures itératives (ou répétitives)



1. L'instruction d'affectation

L'affectation est l'action élémentaire dont l'effet est de donner une valeur à une variable (ranger une valeur à une place).

Syntaxe: Nom_variable $\leftarrow$ Expression;

Exemple :

- $N \leftarrow 10$; /* Cette instruction permet d'affecter la valeur 10 à la variable N */
- $X \leftarrow N$; /* Cette instruction permet d'affecter le contenu de la variable N (10 dans ce cas) à la variable X */
- $S \leftarrow PI * R * R$; /* Cette instruction permet de mettre le résultat du produit dans la variable S */

Remarques:

- ❑ Les deux expressions à droite et à gauche de l'opérateur d'affectation doivent être de même type.
- ❑ L'affectation est une expression comme une autre, c'est-à-dire qu'elle retourne une valeur. Il est donc possible d'écrire: $X \leftarrow Y \leftarrow Z + 2$;



1. L'instruction d'affectation

Exercice d'application 1:

Quelles sont les valeurs successives prises par les variables X et Y suit aux instructions suivantes :

Instruction	Variable X	Variable Y
$X \leftarrow 1$		
$Y \leftarrow -4$		
$X \leftarrow X+3$		
$X \leftarrow Y-5$		
$Y \leftarrow X+2$		
$Y \leftarrow Y-6$		



1. L'instruction d'affectation

Correction :

Instruction	Variable X	Variable Y
$X \leftarrow 1$	1	(vide)
$Y \leftarrow -4$	1	-4
$X \leftarrow X+3$	4	-4
$X \leftarrow Y-5$	-9	-4
$Y \leftarrow X+2$	-9	-7
$Y \leftarrow Y-6$	-9	-13



1. L'instruction d'affectation

Exercice d'application 2: Echanger de deux valeurs

Soit 2 variables quelconques (nombres) x et y ayant respectivement comme valeur -3 et 14. Ecrire un algorithme qui échange le contenu de deux variables ?

Correction : **Algorithme** **Echange;**

Variables x, y, z : entier;

Début

x ← -3 ;	/* initialisation de x */
y ← 14 ;	/* initialisation de y */
z ← x ;	/* on stocke la valeur de x dans z */
x ← y ;	/* on peut maintenant écrire dans x */
y ← z ;	/* on remet l'ancien contenu de x dans y */

Fin.



2. Les instructions d'entrée/sortie

2.1 L'instruction d'entrée (ou de lecture)

Cette instruction permet de prendre une information depuis le clavier et la mettre dans une variable.

Syntaxe

Lire(variable 1, variable 2, variable 3,....., variable n)

Exemple:

Lire(R) ; /* cette instruction permet de saisir le contenu de la variable R */

Lire(x,y) ; /* cette instruction permet de saisir le contenu des variables x et y*/



2. Les instructions d'entrée/sortie

2.2 Instruction de sortie (ou d'écriture)

Cette instruction permet d'afficher à l'écran les valeurs des variables et les messages.

Syntaxe

Ecrire (Expression 1, Expression 2, Expression 3,....., Expression n)

Exemples:

- **Ecrire**(S) ; /* cette instruction permet d'afficher le contenu de la variable S */
- **Ecrire**(x,y) /* cette instruction permet d'afficher le contenu des variables x et y */
- **Ecrire**("Bonjour") /* cette instruction permet d'afficher le message (Bonjour) */
- **Ecrire**("La surface est =", S) ; /* cette instruction permet d'afficher le contenu de S
après l'affichage du message (La surface est =) */

Remarque:

- ❑ Toujours précéder les instructions de lecture par l'affichage d'un message qui invite l'utilisateur à saisir une valeur.



2. Les instructions d'entrée/sortie

Exemple: Algorithme qui échange le contenu de deux variables x et y, en saisissant le contenu des deux variables.

Algorithme Echange;

Variable x, y, z : entier;

Début

Ecrire ("Donner la valeur de x :");

lire(x);

Ecrire ("Donner la valeur de y :");

lire(y);

$z \leftarrow x$;

$x \leftarrow y$;

$y \leftarrow z$;

Ecrire("La valeur de x est :", x) ;

Ecrire("La valeur de y est :", y) ;

Fin.



2. Les instructions d'entrée/sortie

Exercice d'application 3:

Ecrire un algorithme permettant de multiplier deux nombres A et B, en utilisant les instruction de lecture et d'écriture.

Correction :

Algorithme

Multiplication;

Variables A, B, C : entier;

Début

```
Ecrire ("Donner la valeur de A :");  
lire(A);  
Ecrire ("Donner la valeur de B :");  
lire(B);  
C ← A*B ;  
Ecrire("Le produit de ",A, " et ",B, "est", C) ;
```

Fin.

Exécution :

Donner la valeur de A : 5

Donner la valeur de B : -11

le produit de 5 est de -11 est : -55



Exercice 1:

Quelle seront les valeurs des variables A,B et C après exécution des instructions suivantes ?

Algorithme **Affectation1;**

Variables A, B, C : entier;

Début

$A \leftarrow 8 ;$

$B \leftarrow -2 ;$

$C \leftarrow A+B ;$

$A \leftarrow 4 ;$

$C \leftarrow B-A ;$

Fin.



Exercice 2:

Quelle seront les valeurs des variables A et B après exécution des instructions suivantes ?

Algorithme **Affectation2;**

Variables A, B: entier;

Début

$A \leftarrow 2 ;$

$B \leftarrow A+5 ;$

$A \leftarrow A+B ;$

$B \leftarrow B+2 ;$

$A \leftarrow B-A ;$

Fin.



Exercice 3:

Que produit l'algorithme suivant ?

Algorithme

Affectation_ES;

Variables A, B: entier;

Début

Ecrire("Entrez la valeur de A : ") ;

lire(A);

Ecrire("Entrez la valeur de B : ") ;

lire(B);

$A \leftarrow A+B$;

$B \leftarrow A-B$;

$A \leftarrow A-B$;

Ecrire("A= ",A);

Ecrire("B= ",B)

Fin.



Exercice 4:

Que produit l'algorithme suivant ?

Algorithme **Affectation_ES;**

Variables A, B, C : chaine;

Début

Ecrire("Entrez la valeur de A : ") ;

lire(A);

Ecrire("Entrez la valeur de B : ") ;

lire(B);

$C \leftarrow A \&B$;

Ecrire("C= : ",C);

Fin.



Exercice 5:

Ecrire un algorithme qui calcul et affiche la circonférence et la surface d'un disque ($L = 2\pi R$ et $S = \pi R^2$) ? L'algorithme demandera à l'utilisateur d'entrer la valeur du rayon R .

Exercice 6:

Écrire un algorithme qui permet de saisir le prix hors taxe (PHT) d'un article et de calculer son prix total TTC (PTTC), sachant que le TVA=20%.

Exercice 7:

Ecrire un algorithme qui lit le prix hors taxe (PHT), le nombre d'article (NB) et le taux de la valeur ajoutée (TVA), et qui fournit le prix total correspondant (PTTC) ?

Exercice 8:

Ecrire un algorithme qui effectue la lecture t en seconde, et il affiche le temp t en jours, heures, minutes, secondes.

Exemple: si $t = 21020$ secondes, l'algorithme affichera 0 jours 5 heures 50 minutes et 20 secondes.



3. Les structures conditionnelles

3.1 La structure Si Alors.....FinSi

Cette structure est utilisée si on veut **exécuter un bloc d'instruction** seulement si une condition est **vraie** et **ne rien faire** si la condition est **fausse**.

Syntaxe

Si Condition **Alors**

Bloc d'instructions

FinSi

Exemple 1: Ecrire un algorithme qui permet de calculer le résultat de la division de deux entiers A et B ?

3. Les structures conditionnelles



Solution:

Algorithme Division;

Variables A, B, R: entiers;

Début

Ecrire("Entrez le dividende : ") ;

Lire(A);

Ecrire("Entrez le diviseur : ") ;

Lire(B);

Si $B \neq 0$ **Alors**

$R \leftarrow A/B$;

Écrire ("Le résultat de la division est : " , R);

FinSi

Fin.



3. Les structures conditionnelles

Remarques:

- ❑ Une condition est une expression logique ou une variable logique évalué à **Vrai** ou **Faux**.
- ❑ Les instructions seront exécutés si et seulement si la condition est **Vraie**.
- ❑ Une condition formée de plusieurs conditions simples reliées par des opérateurs logiques (**ET**, **OU**, **OU exclusif (XOR)** et **NON**) s'appelle **condition composée**.

Exemples:

- x compris entre 2 et 6 : $(x > 2) \text{ ET } (x < 6)$
- n divisible par 3 ou par 2 : $(n \bmod 3 = 0) \text{ OU } (n \bmod 2 = 0)$
- deux valeurs et deux seulement sont identiques parmi a, b et c : $(a=b) \text{ XOR } (a=c) \text{ XOR } (b=c)$



3. Les structures conditionnelles

- L'évaluation d'une condition composée se fait selon des règles présentées généralement dans ce qu'on appelle **tables de vérité**.

ET logique

C1	C2	C1 ET C2
VRAI	VRAI	VRAI
VRAI	FAUX	FAUX
FAUX	VRAI	FAUX
FAUX	FAUX	FAUX

OU logique

C1	C2	C1 OU C2
VRAI	VRAI	VRAI
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

OU exclusif (XOR)

C1	C2	C1 XOR C2
VRAI	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

NON logique

C1	NON C1
VRAI	FAUX
FAUX	VRAI

Tables de vérité



3. Les structures conditionnelles

Exemple 2: Écrire un algorithme qui teste si une note saisie au clavier est comprise entre 0 et 20.

Solution :

Algorithme TestNote

Variable Note : réel;

Début

Écrire("Entrez la note :");

Lire(Note);

Si Note $\geq$ 0 **ET** Note $\leq$ 20 **Alors**

Écrire ("La note ", Note, "est correcte");

FinSi

Fin



3. Les structures conditionnelles

3.2 La structure Si Alors.....Sinon.....FinSi

Syntaxe

Si Condition **Alors**

Bloc1 d'instructions

Sinon

Bloc2 d'instructions

FinSi

- ❑ Si la condition est **vraie**, le bloc1 d'instructions est exécutée et le bloc2 d'instructions est ignoré, la machine sautera directement à la première instruction située après le **FinSi**.
- ❑ Si la condition est **fausse**, la machine saute directement à la première ligne située après le **Sinon** et exécute le bloc2 d'instructions.



3. Les structures conditionnelles

Exemple 3 : Écrire un algorithme qui affiche si un nombre entier N saisi au clavier est pair ou impair.

Solution : On dit qu'un nombre entier N est pair si le reste de la division entière de N par 2 est égal à zéro. Sinon il est impair.

Algorithme **parité;**

Variable N : entier;

Début

Ecrire(" Donner votre nombre ");

lire(N);

Si (N mod 2) = 0 **Alors**

Ecrire (" ce nombre est paire ");

Sinon

Ecrire (" ce nombre est impaire ");

FinSi

Fin.



3. Les structures conditionnelles

3.3 La structure Si imbriquée (Si ... Alors ... Sinon Si ... Sinon ... Finsi)

Syntaxe

Si Condition1 **Alors**

Bloc1 d'instructions

Sinon

Si Condition2 **Alors**

Bloc2 d'instructions

Sinon

Bloc2 d'instructions

Finsi

Finsi

Note:

L'instruction qui sera exécutée est celle dont la condition est **Vraie**. Si aucune condition n'est vérifiée, c'est l'instruction qui suit le **Sinon** qui sera exécutée.

Exemple 4:

Écrire un algorithme qui permet d'afficher le maximum parmi deux nombres saisis au clavier ?

Solution

Remarque

Algorithme Max

Variables A, B : réels;

Début

 Écrire("Entrez la valeur de A :");

 Lire(A);

 Écrire("Entrez la valeur de B :");

 Lire(B);

Si A > B **Alors**

 Écrire("le maximum est " , A);

Sinon

Si A=B **Alors**

 Écrire("les deux nombres sont égaux ");

Sinon

 Écrire("le maximum est " , B);

Finsi

Finsi

Fin

Une structure conditionnelle peut contenir elle même une autre structure conditionnelle. Dans ce cas, les instructions à exécuter après **Alors** et ou **Sinon** sont à leur tour, des instructions **Si ... Alors . . . Sinon**.



3. Les structures conditionnelles

Exemple 5:

Écrire un programme permettant de calculer des allocations familiales sachant que le montant dépend du nombre d'enfants à charge.

- ❖ Si le nombre d'enfants est inférieur ou égal à 3 alors les allocations sont de 150 euros par enfant.
- ❖ Si le nombre est strictement supérieur à 3 et inférieur à 6 alors les allocations sont de:
 - 150 euros par enfant pour les 3 premiers ;
 - 38 euros pour les suivants.
- ❖ Si le nombre d'enfants est strictement supérieur à 6 alors les allocations sont :
 - 150 euros par enfant pour les 3 premiers ;
 - 38 euros pour les 3 suivants.
 - 0 euros par enfant pour les suivants.

Solution :

Algorithme Allocation_ familiale

Variables NE : réel ; /* NE : Nombre d'enfants */

Variables M : réel ; /* M : montant des allocations */

Début

Écrire("Entrez le nombre d'enfants :");

Lire(NE);

Si NE ≤ 3 **Alors**

M ← NE * 150;

Sinon

Si NE ≤ 6 **alors**

 M ← (3 * 150) + (NE - 3) * 38;

Sinon

 M ← (3 * 150) + (3 * 38);

Finsi

Finsi

Écrire("Le montant des allocations est ", M)

Fin



3. Les structures conditionnelles

3.4 La structure à choix multiple (Selon.....faire.....Sinon.....Fin)

Cette instruction permet de faire le choix entre plusieurs alternatives.

Syntaxe

Selon Variable **Faire**

Val1: instruction1;

Val2: instruction2;

Val3: instruction3;

.....

.....

ValN: instructionN;

Sinon

instruction par défaut;

FinSelon



3. Les structures conditionnelles

- ❑ Lorsque l'ordinateur rencontre cette instruction, il vérifie la valeur de la variable et il la compare aux différentes valeurs.
- ❑ Les valeurs sont évaluées dans l'ordre, les unes après les autres , et dès qu'une de celles-ci est vérifiée l'action associée est exécutée.
- ❑ On peut utiliser une instruction **Sinon** (facultative), dont l'action sera exécutée si aucune des valeurs évaluées n'est remplie.

Exemple 6: Écrire un algorithme permettant d'afficher le jour en toute lettre selon son numéro saisi au clavier ?

Solution :

Algorithme jour;
Variable j : entier;

Début

lire(j);

Selon j Faire

1: Ecrire(" Dimanche ");

2: Ecrire (" Lundi ");

3: Ecrire (" Mardi ");

4: Ecrire(" Mercredi ");

5: Ecrire(" Jeudi ");

6: Ecrire(" Vendredi ");

7: Ecrire(" Samedi ");

Sinon

Ecrire(" Erreur ");

FinSelon

Fin.



TD2: Les structures conditionnelles

Exercice 1:

Ecrire un algorithme qui lit deux entiers A et B, et affiche le plus grand ?

Exercice 2:

Ecrire un algorithme qui lit trois entiers A, B et C, et affiche le plus grand ?

Exercice 3:

Un magasin accorde à ses clients, une réduction de 5% pour les montants d'achats supérieurs à 150 euros.

Écrire un algorithme permettant de saisir le prix total hors taxe (PHT) et de calculer le montant toute taxe (PTTC) en prenant compte de la remise et de la TVA à 20%.

Exercice 4:

Ecrire un algorithme qui donne l'état de l'eau selon sa température.

- Si $T < 0$: état solide.
- Si $T > 100$: état gazeux.
- Si $0 < T < 100$: état liquide.



Exercice 5:

Ecrire un algorithme qui donne la mention selon la moyenne générale obtenue par un élève.

- Si moyenne < 10 : Insuffisant
- Si $10 < \text{moyenne} < 12$: Passable
- Si $12 < \text{moyenne} < 14$: Assez bien
- Si $14 < \text{moyenne} < 16$: Bien
- Si moyenne > 16 : Très bien

Exercice 6:

Ecrire un algorithme qui affiche selon un numéro compris entre 1 et 12 le mois équivalent ?



4. Les structures itératives (ou répétitives)

- ❑ Les structures itératives (ou répétitives) constituent des boucles qui répètent l'exécution d'un même bloc d'instructions un certain nombre de fois.

- ❑ Il existe trois types de boucles :
 - Boucle «Tant que..... Faire »
 - Boucle «Répéter.....jusqu'à»
 - Boucle «Pour»



4. Les structures itératives (ou répétitives)

4.1 La boucle: Tant que Faire

Cette boucle est utilisée pour exécuter des actions tant qu'une condition est **Vraie**.

Syntaxe:

Tant que Condition **Faire**

Actions;

FinTantque

- ❑ L'exécution de la boucle dépend de la condition. Si elle est **vraie**, le programme exécute les instructions qui suivent, jusqu'à ce qu'il rencontre la ligne **FinTantque**, puis il retourne ensuite sur la ligne du **Tantque**, et procède au même examen, et ainsi de suite.
- ❑ La boucle ne s'arrête que lorsque la condition devient **fausse**, et dans ce cas le programme poursuit son exécution après **FinTantque**.



4. Les structures itératives (ou répétitives)

Exemple 1: Donner les valeurs de N après l'exécution de l'algorithme ci-dessous ?

Algorithme affiche_nbrs;

Nous avons saisis par exemple la valeur 9

Variable N : entier;

Le résultat de l'algorithme après

Début

l'Exécution :

Ecrire(" Entrer la valeur de N");

Lire(N);

Tant que (N >= 0) **faire**

 Ecrire(N) ;

 Ecrire("\n"); /* Retour à la ligne */

 N ← N - 1 ;

FinTantque

Fin.

9
8
7
6
5
4
3
2
1
0



4. Les structures itératives (ou répétitives)

Exemple 2

Ecrire un algorithme qui demande un entier de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27?

Solution

Algorithme affiche_nbrs;

Variables N,i : entiers;

Début

Ecrire("Entrer la valeur de N");

Lire(N);

$i \leftarrow 1$;

Tant que (i <= 10) **faire**

$N \leftarrow N+1$;

Ecrire(N) ;

$i \leftarrow i+1$;

FinTantque

Fin.



4. Les structures itératives (ou répétitives)

4.2 La boucle: Répéter.....jusqu'à

Cette instruction est utilisée pour exécuter des actions jusqu'à ce qu'une condition soit remplie (**Vraie**).

Syntaxe:

Répéter

Actions;

Jusqu'à Condition

Remarque:

❑ La vérification de la condition s'effectue après les actions. Celles-ci sont donc exécutées au moins une fois.



4. Les structures itératives (ou répétitives)

Exemple 1:

Donner les valeurs de y après l'exécution de l'algorithme ci-contre ?

Exécution :

$y=1$

$y=3$

$y=6$

$y=10$

$y=15$

$y=21$

$y=28$

Algorithme

bloc;

Variable

i, y : entier;

Début

$i \leftarrow 0;$

$y \leftarrow 0;$

Répéter

$i \leftarrow i + 1 ;$

$y \leftarrow y + i ;$

Ecrire("y=", y);

Ecrire("\\n");

Jusqu'à ($i=7$)

Fin.



4. Les structures itératives (ou répétitives)

Exemple 2: Ecrire un algorithme qui calcule la somme des N premiers nombres entiers ?
On suppose que N est strictement positif.

Solution:

Algorithme **Somme**

Variables S, i, N : entiers

Début

Écrire(" Entrez une valeur strictement positive : ");

Lire(N);

$S \leftarrow 0;$

$i \leftarrow 0;$

Répéter

$i \leftarrow i + 1;$

$S \leftarrow S + i;$

Jusqu'à $i \geq N$

Écrire("La somme des nombres ", N, "premiers entiers est :", S)

Fin



4. Les structures itératives (ou répétitives)

4.3 La boucle: Pour.....Faire

La spécification de cette instruction c'est qu'elle limite le nombre de fois où le bloc Actions (ou instructions) doit se répéter.

Syntaxe: **Pour** compteur $\leftarrow$ *Valeur_initiale* **jusqu'à** *Valeur_finale* **Faire**

Actions;

FinPour

- ❑ La variable **compteur** est de type **entier**. Elle est initialisée à la valeur initiale. Le compteur augmente sa valeur de 1 automatiquement à chaque tour de boucle jusqu'à la valeur finale.
- ❑ Pour chaque valeur prise par la variable compteur, la liste des **Actions** est exécutée.
- ❑ Lorsque la variable compteur vaut la valeur finale, le traitement est exécuté une dernière fois puis le programme sort de la boucle.



4. Les structures itératives (ou répétitives)

Exemple 1: Donner les valeurs de *i* après l'exécution de l'algorithme ci-dessous ?

Algorithme Afficher_chiffre;

Variable *i* : entier;

Début

Pour *i*←1 **Jusqu'à** 7 **Faire**

 Ecrire(*i*);

 Ecrire("\n"); /* Retour à la ligne */

Finpour

Fin.

Exécution :

1

2

3

4

5

6

7



4. Les structures itératives (ou répétitives)

Exemple 2: Ecrire un algorithme qui affiche à l'écran la table de multiplication de 7 ?

Solution:

Algorithme Table_de_matière_de7;

Variable k : entier;

Début

Pour k $\leftarrow$ 0 **Jusqu'à** 10 **Faire**

Ecrire("7* ",k,"=", 7*k);

Ecrire("\n");

FinPour

Fin.



4. Les structures itératives (ou répétitives)

Exemple 3: Ecrire un algorithme qui calcule la somme S des 10 premiers entiers naturels

$$S = 1+2+3+\dots +10 ?$$

Solution:

Algorithme **Somme;**

Variables S,i : entiers;

Début

$S \leftarrow 0;$

Pour i $\leftarrow$ 0 **Jusqu'à** 10 **Faire**

$S \leftarrow S + i;$

FinPour

 Écrire("La somme des ", 10, "premiers entiers est :", S);

Fin.



4. Les structures itératives (ou répétitives)

Remarques:

- ❑ Par défaut la variable compteur est incrémentée de 1 à chaque tour de boucle.
- ❑ Pour modifier la valeur d'incrémentation, il suffit de rajouter le mot **Pas** et la valeur de ce pas à la boucle.

Syntaxe:

Pour Compteur $\leftarrow$ *Valeur_initiale* **jusqu'à** *Valeur_finale* **Pas** ValeurPas **Faire**
Instructions;
FinPour

Exemple 4: Écrire un algorithme qui permet de saisir un entier et qui calcule la somme S des entiers pairs compris entre zéro et ce nombre ?



4. Les structures itératives (ou répétitives)

Solution:

Algorithme Somme_Paire

Variables S, i, N : entiers

Début

Écrire(" Entrez la valeur de N : ");

Lire(N);

$S \leftarrow 0;$

Pour $i \leftarrow 0$ jusqu'à N **Pas 2 Faire**

$S \leftarrow S + i;$

FinPour

Écrire("La somme des nombres pairs est :", S);

Fin



4. Les structures itératives (ou répétitives)

Remarques:

- ❑ On utilise la boucle « **Pour** » quand on connaît le nombre d'itérations à l'avance.
- ❑ Les boucles « **Répéter** » et « **TantQue** » sont utilisées lorsqu'on ne sait pas au départ combien de fois il faudra exécuter ces boucles.
- ❑ A la différence de la boucle « **Tantque** », la boucle « **Répéter** » est exécutée au moins une fois.
- ❑ La condition d'arrêt de la boucle « **Répéter** » est la négation de la condition de poursuite de la boucle « **Tantque** ».



Exercice 1:

1. Écrivez un algorithme qui affiche 100 fois la phrase : "je suis un physicien".
2. Écrivez un algorithme qui affiche les entiers de 1 à 100.
3. Écrivez un algorithme qui affiche les entiers pairs de 0 à 100.

Exercice 2:

Ecrire un algorithme qui calcule le Plus Grand Commun Diviseur "PGCD" de deux entiers à l'aide de l'algorithme d'Euclide (exemple $A = 12$ et $B = 8$, Alors $\text{PGCD} = 4$).

Exercice 3 :

Ecrire un algorithme qui calcul x à la puissance n où x est un réel non nul et n un entier positif ou nul ?

TD3: Les structures itératives (ou répétitives)



Exercice 4:

Ecrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 10 nombres :

Entrez le nombre numéro 1 : 7

Entrez le nombre numéro 2 : 24

.....etc.

Entrez le nombre numéro 10 : 13

Le plus grand de ces nombres est : 24

Exercice 5:

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle ?

NB : la factorielle de 8, notée $8!$, vaut $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$

Exercice 6:

Écrire un algorithme qui demande 10 entiers, compte le nombre d'entiers positifs saisis, et affiche ce résultat ?