

# Chapitre 4

## Les structures de contrôle

# Exécution non séquentielle

- **Les branchements conditionnels**
  - if else
  - Switch
- **Les boucles**
  - while
  - do while
  - for
- **Les branchements inconditionnels**
  - break
  - continue
  - goto
  - return
  - exit

- Traduction des instructions : schéma conditionnel

Syntaxe en algo	Syntaxe en C
Si condition Alors Début Si instructions Fin Si	if ( condition ) { instructions }
Si condition Alors Début Si instructions Fin Si Sinon Début Sinon instructions Fin Sinon	if ( condition ) { instructions } else { instructions }

- Exercice

Traduire en C

```
Si X > 0 alors
Début Si
    Ecrire("X supérieur à 0")
Fin si
Sinon
Début Sinon
    Si X=0 alors
    Début Si
        Ecrire("X égal à 0")
    Fin Si
    Sinon
    Début Sinon
        Ecrire("X inférieur à 0")
    Fin Sinon
FinSinon
```

- **Corrigé**  
Traduire en C

```
if (X > 0)
{
    printf("X supérieur à 0") ;
}
else
{
    if (X==0)
    {
        printf("X égal à 0") ;
    }
    else
    {
        printf("X inférieur à 0") ;
    }
}
```

# Branchements conditionnels (Switch)

- **Switch** (brancher)

```
switch(expression)  
{  
    case cst1: instr1;  
    case cst2: instr2;  
    ...  
    default:instrF;  
}
```

L'expression (entière) est évaluée. On passe alors au case correspondant et on exécute les instructions à partir de ce point. Si aucun case ne correspond, on exécute instrF.

```
switch(c){  
    case 'a':    printf("aba");  
    case 'e':    printf("ebe");  
    default:    printf("ibi");  
}
```



# Choix multiple: « switch case »

```
/* Utilisation de switch case */
main()
{
    char choix;
    ...
    switch(choix)
    {
        case 'a' : fonctionA();
        case 'b' : fonctionB();
        case 'c' : fonctionC();
        default : erreur(3);
    }
}
```

Paramètre de décision

Exécuté si choix = a

Exécuté si choix = a ou b

Exécuté si choix = a, b ou c

Exécuté si choix non répertorié par un « case »

# Effet du « break »

```
/* Utilisation de switch case */
```

```
main()
```

```
{
```

```
    char choix;
```

```
    ...
```

```
    switch(choix)
```

```
    {
```

```
        case 'a' : fonctionA(); break
```

```
        case 'b' : fonctionB(); break
```

```
        case 'c' : fonctionC(); break;
```

```
        default : erreur(3);
```

```
    }
```

```
}
```

Paramètre de décision

Exécuté si choix = a

Exécuté si choix = b

Exécuté si choix = c

Exécuté si choix non répertorié par un « case »



- Traduction des instructions : boucle Pour

Syntaxe en algo	Syntaxe en C
Pour $i \leftarrow$ «valeur initiale» à « valeur finale» [de pas p ] Faire instructions Fin Pour	for ( $i =$ « valeur initiale» ; $i \leq$ « valeur finale» ; $i = i + p$ ) { instructions }

Syntaxe en algo	Syntaxe en C
Pour $i \leftarrow 1$ à 100 de pas 1 Faire ... Fin Pour	for ( $i = 1$ ; $i \leq 100$ ; $i = i + 1$ ) { ... }
Pour $i \leftarrow 1$ à 100 Faire ... Fin Pour	for ( $i = 1$ ; $i \leq 100$ ; $i++$ ) { ... }

- Exercice

Traduire en C

Algorithme Factorielle

Variables n, fact : Entier

Début

fact  $\leftarrow$  1

Pour n  $\leftarrow$  1 à 100 Faire

fact  $\leftarrow$  (n \* fact)

Fin pour

Ecrire ("La factorielle de 100 est ", fact)

Fin

- Corrigé

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

int main ( )
{
    int n, fact;
    fact = 1 ;
    for (n=1; n<=100; n++)
    {
        fact = (n * fact) ;
    }
    printf ("La factorielle de 100 est %d", fact) ;

    return EXIT_SUCCESS;
}
```

- Traduction des instructions : boucle Tant que et répéter tant que

Syntaxe en algo	Syntaxe en C
Tant que condition Faire instructions Fin Tant que	while (condition) { instructions }

Syntaxe en algo	Syntaxe en C
Répéter instructions Tant que condition	do { instructions }while (condition);

- Exercice

Traduire en C

Algorithme Saisie

Variable n : Entier

Début

Lire (n)

Tant que  $n > 0$  Faire

Écrire ("Saisissez un nombre")

Lire(n)

Fin Tant que

Fin

- **Corrigé**

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int n ;
    scanf("%d",&n);
    while (n > 0)
    {
        printf("Saisissez un nombre");
        scanf("%d",n);
    }
    return EXIT_SUCCESS;
}
```



# Exemples

- **Exemple 1**

*/\* Afficher les nombres de 0 à 9 \*/*

```
int I = 0;
while (I < 10) {
    printf("%i \n", I); I++; }
```

## **Exemple 2**

*/\* Afficher les nombres de 0 à 9 \*/*

```
int I;
I = 0;
while (I < 10)
    printf("%i \n", I++);
```

*/\* Afficher les nombres de 1 à 10 \*/*

```
I = 0;
while (I < 10)
    printf("%i \n", ++I);
```

# Branchements inconditionnels

- **break**: provoque la sortie immédiate de la boucle ou du switch en cours.
- **continue**: passage à la prochaine itération d'une boucle.
- **goto**: branchement sur une ligne labellisée.
- **return**: permet de sortir (proprement) de la fonction et de retourner une valeur (résultat de la fonction,...)
- **exit**: permet de quitter le programme avec une valeur, avec flush et fermeture des fichiers, libération de la mémoire....

# Travaux pratiques

---

- Exercice

Ecrire un programme qui permet de calculer et d'afficher le produit deux entiers saisis au clavier

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

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

```
int main(){
```

```
    int x, y, prod;
```

```
    printf("Entrer deux valeurs entiers :");
```

```
    scanf("%d %d",&x, &y);
```

```
    prod = x*y;
```

```
    printf("Le produit de %d et de %d est : %d", x, y, prod);
```

```
    return EXIT_SUCCESS;
```

```
}
```

# Travaux pratiques

---

- Exercice

Ecrire un programme qui permet de calculer et d'afficher la somme de deux flottants saisis au clavier

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

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

```
int main(){
```

```
    float x, y, som;
```

```
    printf(" Entrer deux valeurs entiers :");
```

```
    scanf("%f %f",&x, &y);
```

```
    som = x+y;
```

```
    printf("le produit de %f et de %f est : %f", x, y, som);
```

```
    return EXIT_SUCCESS;
```

```
}
```

# Travaux pratiques

---

- Exercice

Ecrire un programme qui permet de comparer deux entiers :

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

int main(){
    int x, y ;
    printf("Entrer deux entiers ");
    scanf("%d %d", &x, &y);
    if (x > y){
        printf("%d est supérieur à %d", x, y);
    }
    else {
        printf("%d est supérieur à %d", y, x);
    }
    return EXIT_SUCCESS;
}
```



# Exercices

- **Exercice 1:** Saisir un caractère au clavier avec scanf. Afficher son code ASCII à l'écran

- **Solution 1**

```
#include <stdio.h>
void main() {
    char caractere ;
    scanf ("%c ", &caractere );
    printf ("dode ASCII du caractère saisi :%d ", caractere );
    return 0;
}
```



# Exercices

- **Exercice 2**

- Ecrire un programme C permettant de poser la question 'voulez-vous jouer ?'
- Il y aura saisie d'une réponse avec la fonction **getchar**. Si la réponse est 'o' ou 'O' vous affichez " c' est parti ! ". Si la réponse est 'n' ou 'N' vous affichez " tant pis"

# Exercices

- **Solution 2**

```
#include <stdio.h>
void main() {
    char reponse ;
    printf ("Voulez-vous jouer ?");
    reponse=getchar() ;
    if (reponse=='o' || reponse=='O')
        printf ("\n c'est parti");
    else if (reponse=='n' || reponse=='N')
        printf ("\n tant pis");
}
```

# Exercices

- **Exercice 3:** Saisir un entier  $n$  puis calculer  $n$  !
  - 1) Utiliser une boucle **while**
  - 2) Utiliser une boucle **for**

# Exercices

- **Solution 3**

*/\*solution avec for\*/*

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

```
void main() {
```

```
    int  i, n, result;
```

```
    scanf("%d ", &n) ; /*saisi l'entier n*/
```

```
    for(i=1 ; i<=n ; i++)
```

```
        result=result*i;           /*calcul de la factorielle*/
```

```
    printf(" %d ", result); /* affichage*/
```

```
}
```

# Exercices

- **Solution 3 suite**

```
/*solution avec while*/
#include <stdio.h>
void main() {
    int i, n, result;
    scanf("%d ", &n) ; /*saisi l'entier n*/
    while(i<=n)
    {
        result=result*i;      /*calcul de la factorielle*/
        i++ ;
    }

    printf(" %d ", result); /* affichage*/
}
```



# Exercices

- **Exercice 4:** Ecrire un programme qui affiche la moyenne d'une suite d'entiers positifs entrés au clavier. On arrêtera la saisie quand le nombre -1 est entré, comme dans l'exemple suivant :

Entrez un entier positif : 5

Entrez un entier positif : 2

Entrez un entier positif : 3

Entrez un entier positif : -1

La moyenne de ces 3 entiers vaut 3.333333



# Exercices

- **Solution 4 4:**

```
/******  
/** calcul de la moyenne d'une suite d'entiers */  
#include <stdio.h>  
main() {  
    int n = 0, x = 0, somme = 0;  
    do  
    {  
        somme += x;  
        printf("Entrez un entier positif : ");  
        scanf("%d",&x);  
        n++;  
    }  
    while (x != -1);  
    printf("La moyenne de ces %d entiers vaut %f\n",n - 1, (float)somme/(n-1));  
}
```

# Exercices

- **Exercice 5:** Ecrire un programme qui calcule  $x^n$  où  $x$  est un nombre réel de type **double** et  $n$  un **entier**, tous deux **entrés au clavier**. On écrira le programme en utilisant une boucle **for**, puis une boucle **while**

# Exercices

- **Solution 5:**

```
        /**          calcule de x^n          ***/  
#include <stdio.h>  
main() {  
    int n, i;  
    double x, puissance;  
    printf("Entrez x : ");  
    scanf("%lf",&x);  
    printf("Entrez n : ");  
    scanf("%d",&n);  
    printf("\n Calcul de x^n avec une boucle for\n");  
    for (i = 1, puissance = 1; i <= n; i++)  
        puissance *= x;  
    printf("(%lf) ^ %d = %lf\n",x,n,puissance);  
    printf("\n Calcul de x^n avec une boucle while\n");  
    i = 0;  puissance = 1;  
    while (i < n) {  
        puissance *= x;  
        i++;  
    }  
    printf("(%lf) ^ %d = %lf\n",x,n,puissance);  
}
```