

Révision Générale – Structures de Contrôle et Données

E LEARNING BTS

Séance du 18 Septembre 2025

1 Les Boucles

1.1 La Boucle Tant Que

Boucle Tant Que – Algorithmique

Syntaxe Algorithmique :

```
tant que condition faire
    instruction
fin tant que
```

Description : La condition est testée avant l'exécution des instructions. Si elle est vraie, les instructions sont exécutées, sinon la boucle s'arrête.

Boucle Tant Que – Langage C

Syntaxe en C :

```
while (condition) {
    instruction ;
}
```

Exemple :

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

1.2 La Boucle Faire - Tant Que

Boucle Faire - Tant Que – Algorithmique

Syntaxe Algorithmique :

```
faire
    instruction
tant que condition
```

Description : Les instructions sont exécutées au moins une fois, puis la condition est testée. Si elle est vraie, la boucle continue.

Boucle Faire - Tant Que – Langage C

Syntaxe en C :

```
do {  
    instruction ;  
} while (condition);
```

Exemple :

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

1.3 La Boucle Pour

Boucle Pour – Algorithmique

Syntaxe Algorithmique :

```
pour indice allant de VD à VF faire  
    instruction  
fin pour
```

Description : Répète les instructions pour chaque valeur de l'indice allant de la valeur de départ (VD) à la valeur finale (VF).

Boucle Pour – Langage C

Syntaxe en C :

```
for (initialisation ; condition ; évolution) {  
    instruction ;  
}
```

Exemple :

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

2 Les Tableaux

Définition

Un **tableau** est une collection d'objets de données de même type occupant des espaces mémoires contigus repérés par des indices.

Déclaration d'un Tableau Vecteur

Algorithmique :

nom_tableau : tableau[taille] de type

Langage C :

```
type nom_tableau[taille];
```

Langage PHP :

```
$nom_tableau = array();
```

Algorithmes de Manipulation de Tableaux

Les opérations courantes sur les tableaux :

- **Stockage** des éléments dans le tableau
- **Affichage** des éléments
- **Rechercher** une valeur
- **Déterminer** les valeurs min, max, moyenne
- **Algorithmes de tri** (Bulle, Permutation, Sélection, ...)

3 Les Procédures et les Fonctions

3.1 Les Procédures

Déclaration d'une Procédure – Algorithmique

Syntaxe :

```
procédure nom_procédure (liste arguments)
déclaration
    variables locales
début
    traitement
fin nom_procédure
```

Appel de la procédure :

```
nom_procédure (liste des paramètres effectifs)
```

Procédure – Langage C

Syntaxe en C :

```
void nom_procedure (liste arguments) {  
    variables locales ;  
    traitement ;  
}
```

Procédure – Langage PHP

Syntaxe en PHP :

```
function nom_procedure (liste arguments) {  
    traitement ;  
}
```

3.2 Les Fonctions

Déclaration d'une Fonction – Algorithmique

Syntaxe :

```
fonction nom_fonction (liste arguments) : type  
déclaration  
    variables locales  
début  
    traitement  
    retour expression  
fin nom_fonction
```

Appel d'une fonction :

```
nom_variable <- nom_fonction (liste paramètres effectifs)
```

Description : Une fonction retourne un seul résultat.

Fonction – Langage C

Syntaxe en C :

```
type nom_fonction (liste arguments) {  
    variables locales ;  
    traitement ;  
    return expression;  
}
```

Fonction – Langage PHP

Syntaxe en PHP :

```
function nom_fonction (liste arguments) {  
    traitement ;  
    return expression;  
}
```

4 Les Fichiers

Définition

Un **fichier** est une structure informatique qui permet le stockage des données d'une façon permanente sur un support externe à la mémoire centrale (MC).
Un fichier peut être de deux types : **texte** ou **binaire**.

Déclaration d'un Fichier

Algorithmique :

nom_fichier : fichier

Langage C :

FILE * nom_fichier;

4.1 Opérations sur les Fichiers

Ouverture d'un Fichier

Algorithmique :

nom_fichier <- ouvrir (nom_physique, mode)

Mode : lecture, écriture, ajout (à la fin)

Langage C :

nom_fichier = fopen (nom_physique, mode);

Mode = "r" (read), "w" (write), "a" (append)

Fermeture d'un Fichier

Algorithmique :

fermer (nom_fichier)

Langage C :

fclose (nom_fichier);

Lecture d'une Donnée

Algorithmique :

lire (nom_fichier, nom_variable)

Langage C :

- fgetc – lire un caractère
- fgets – lire une chaîne
- fscanf – lecture formatée
- fread – lecture binaire

Écriture d'une Donnée

Algorithmique :

ecrire (nom_fichier, nom_variable)

Langage C :

- fputc – écrire un caractère
- fputs – écrire une chaîne
- fprintf – écriture formatée
- fwrite – écriture binaire

Tester la Fin du Fichier

Algorithmique :

fin_fichier (nom_fichier)

Langage C :

feof(nom_fichier)