

Séance 5

Les sous-programmes

Objectifs

- Comprendre et utiliser le concept de sous-programme

Contenu

1. Déclaration de sous-programme
2. Utilisation de sous-programme

Introduction

- Un sous-programme est une séquence d'instructions ordonnées pour exécuter un traitement spécifique non élémentaire ou faire un calcul,
- Il n'est exécuté dans l'algorithme uniquement qu'à son appel, si nécessaire.

1. Déclaration de sous-programme

1.1. Procédure

1 / 2

6

→ Est un sous-programme qui permet d'effectuer des traitements sur zéro, une ou plusieurs variable(s) et renvoyer les résultats au programme appelant. Par exemple : lecture, affichage, calcul et changement des valeurs des variables,

→ **Déclaration**

procédure <nom_procédure>(<liste des paramètres>)
 < déclaration des objets locaux de la procédure>

Début

 {corps de la procedure}

Fin

Cette ligne est appelée
signature ou entête de la
procédure

1.2. Fonction

2/2

7

→ est un sous-programme qui permet de résoudre un problème précis (généralement un calcul) et renvoie un seul résultat de type simple (entier, réel, logique, caractère ou chaîne). La valeur retournée (calculée) est rendue disponible par l'intermédiaire du nom de la fonction

→ Déclaration

fonction <nom_fonction> (<liste des paramètres>) : <type de résultat>
<déclaration des objets locaux à la fonction>

Début

{corps de la fonction}

retourner(résultat)

Fin

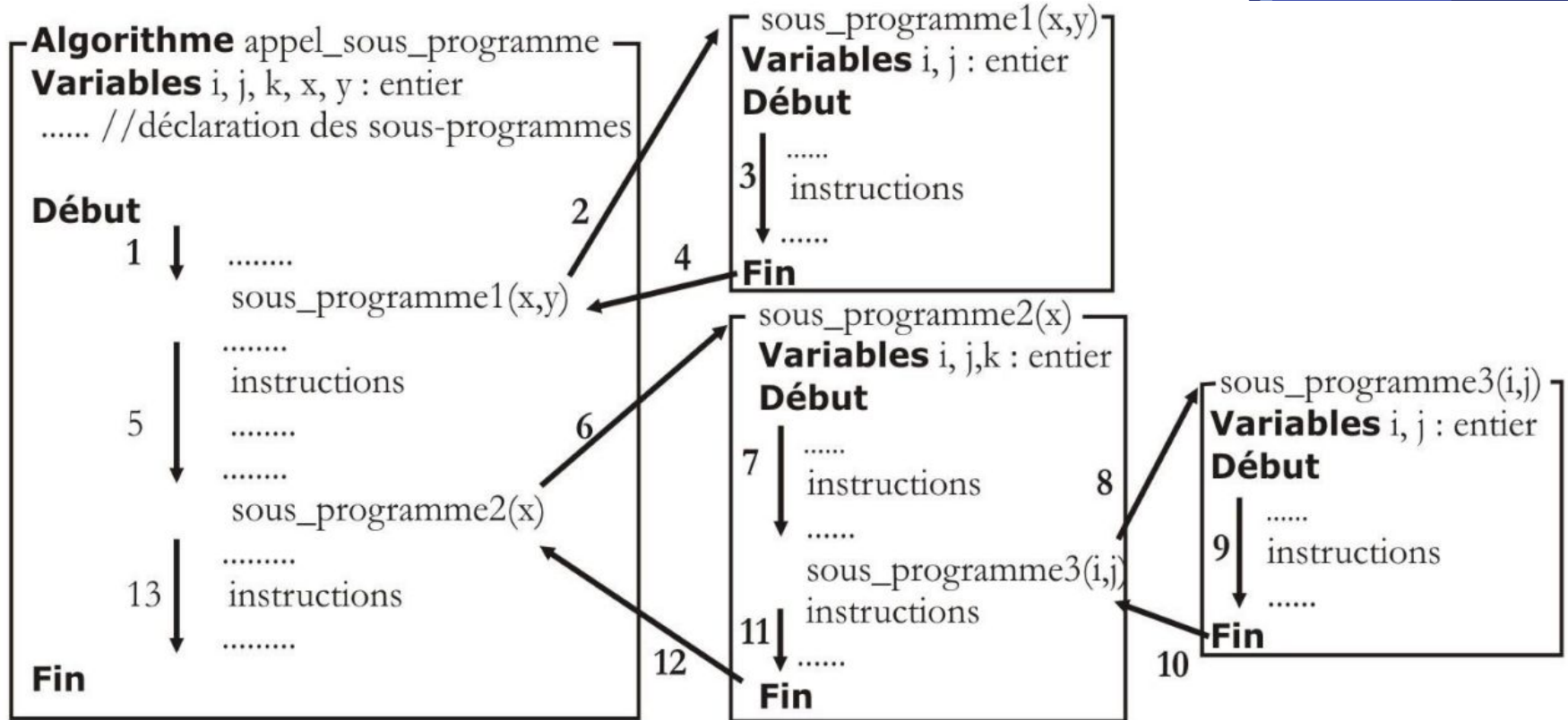
Cette ligne est appelée signature ou entête de la fonction

2. Utilisation de sous-programme

2.1. Appel

- Les sous-programmes peuvent être appelés (ou invoqués pour être exécutés) à partir de l'algorithme principal ou depuis un autre sous-programme.
- Syntaxe :
nom-sous-programme(para-1,para-2,...,para-n)

2.1. Appel



2.2. Portée d'une variable

- La portée d'une variable désigne l'ensemble des sous-programmes où cette variable va être accessible.
- **Variable globale** : Accessible depuis tous les sous-programmes et accessible depuis l'algorithme principal.
- **Variable locale** : Accessible uniquement depuis le sous-programme dans lequel elle est déclarée.

3. Application

Exercice 1

On se propose d'écrire un algorithme qui permet de résoudre une équation du second degré.

1. Écrire une fonction qui prend en entrée les paramètres a , b et c de type réel et renvoie le discriminant.
2. En utilisant cette fonction (que l'on supposera comme disponible), écrire l'algorithme de lit les réels a , b et c à l'écran et résout l'équation du second degré correspondant.

Exercice 2

1. Écrire une procédure qui permet de saisir les valeurs d'une matrice de dimensions (5, 5).
2. Écrire une procédure qui permet d'afficher une matrice dimensions (5, 5).
3. En utilisant les deux précédentes procédures, écrire un algorithme qui permet de :
 - a. saisir les valeurs de deux matrices `matrice1` et `matrice2`,
 - b. calculer la somme des deux matrices `matrice1` et `matrice2`,
 - c. afficher les matrices `matrice1` et `matrice2` ainsi que la matrice somme.

- [1] Y. A.. GBEDEVI, « Initiation à l'algorithmique », Université de Lomé, Support de cours, 2022-2023.
- [2] R. Christophe, « Bases d'algorithmique. Support de Cours au Lycée Vincent d'Indy ». 2015-2016.
- [3] L, Baba Ahmed et S, Hocine, algorithmique et structure de données statistiques. OPU, 2016.
- [4] E. Thiel, « Support de cours Algorithmes et programmation en Pascal ». Faculté des Sciences de Luminy, Université d'Aix-Marseille AMU, 1997.
- [5] A. Rabia, F. Rachid, A. O. Mohand, B. Moufida, et Y. Smain, « Algorithmique :Cours et Exercices en Programmation Pascal ». Cours, Exercices et Programmation Pascal Première Année Universitaire, USTHB, 2018-2017