

*Objectif de la séance :*

Écrire un programme sous forme de fonctions et procédures en pseudo-code.

## 1 Écriture de pseudo-code

Nous reprenons dans un premier temps un exercice directement réalisés en Pascal dans le cours I1.

### *Exercices*

Choisir l'instruction itérative la plus adaptée et écrire l'algorithme pour :

1. en fonction d'un entier  $n$  fourni en entrée, afficher la suite de nombre jusqu'à 0 (exclu), que le chiffre soit positif ou négatif. Exemples :  
Si  $n = 5$ , le programme affiche 5 4 3 2 1  
Si  $n = -3$ , le programme affiche -3 -2 -1.  
Si  $n = 0$ , le programme n'affiche rien.

## 2 Analyse de texte

L'objectif du programme à concevoir est de retrouver au sein d'un texte toutes les apparitions d'un mot donné et d'afficher à quelle(s) position(s) il apparaît.

Le programme demandera d'abord à l'utilisateur de saisir un texte. Ce texte peut contenir plusieurs phrases, des signes de ponctuation, mais pas de caractère numérique (e.g. '0'). Ensuite, le programme demande de saisir un mot (dont le nombre de caractères est compris entre 3 et 10 lettres, minuscules ou majuscules).

Le programme devra rechercher toutes les occurrences de ce mot dans le texte et afficher à l'écran la position dans la chaîne de caractère.

### 2.1 Conception globale

A partir des spécifications données ci-dessus, vous devez réaliser la conception globale du programme, selon le paradigme de la programmation structurée, en définissant une analyse descendante puis les signatures des fonctions et procédures à écrire.

**Remarque no1 :** Une chaîne de caractère est représentée numériquement par un tableau de caractères. Ainsi pour désigner le caractère situé à la position  $i$  d'une chaîne  $str$ , on écrit  $str[i]$ . Le premier indice d'une chaîne est 1.

**Remarque no2 :** Vous pouvez vous appuyer pour l'écriture de ce programme sur des fonctions et procédures existants dans tout langage de programmation impérative pour la manipulation de chaînes de caractères. Il s'agit de :

- fonction *longueur*(*str* : Chaîne de caractères) : *Naturel* qui retourne le nombre de caractères de la chaîne de caractères donnée en paramètre d'entrée ;
- fonction *minuscule*(*texte1* : Chaîne de caractères) : *Chaîne de caractères* qui renvoie le texte passé en entrée en minuscules.

Ces fonctions et procédures existent. Vous n'avez donc pas à écrire leur pseudo-code. Par contre il est nécessaire de les faire apparaître dans l'analyse descendante pour définir les sous-programme où elles seront utilisées.

Pour effectuer cette analyse descendante, nous procéderons suivant les quatre étapes ci-dessous :

---

### *Exercices*

---

1. Définissez de manière informelle les tâches à faire accomplir par le programme, et éventuellement des sous-tâches au sein de chaque tâche.
  2. Rédigez sous la forme d'un diagramme d'analyse descendante cette décomposition fonctionnelle.
  3. Identifiez les entrées et les sorties de chaque sous-programme.
  4. Choisissez de représenter chaque sous-programme par une fonction ou une procédure et écrivez leur signature.
- 

## 2.2 Conception détaillée

---

### *Exercices*

---

1. Écrivez la conception détaillée des fonctions
  2. Écrivez la conception détaillée d'un programme principal
-