

Algorithmique 2

Les Chaines de Caractères

Tarek Boutefara

2026-02-07

Table of Contents

1. Introduction	1
2. Définition et Propriétés	2
2.1. Définition	2
2.2. Propriétés	2
2.3. Accès et Modification d'un caractère	2
3. Caractères spéciaux	4
4. Fonctions et Procédures Standards de Manipulation	5
4.1. La Fonction Longueur (Length)	5
4.2. La Fonction de Concaténation (Concat)	5
4.3. La Fonction d'Extraction (Copy)	5
4.4. La Fonction de Recherche (Pos)	5
4.5. La Procédure de Suppression (Delete)	6
4.6. La Procédure d'Insertion (Insert)	6
5. Exemples d'application	7
5.1. Le Matricule des Véhicules	7
5.2. Le Code-barres (EAN-13)	7
5.3. L'URL (Uniform Resource Locator)	8
5.4. Le Relevé d'Identité Bancaire (RIB)	8
6. Conclusion	9

Chapitre 1. Introduction

Jusqu'à présent, nous avons appris à faire parler l'ordinateur le langage des mathématiques : nous avons manipulé des Entiers pour compter et des Réels pour mesurer. Cependant, le monde réel ne s'écrit pas uniquement avec des chiffres. Pour qu'un programme soit réellement utile à un être humain, il doit pouvoir manipuler des noms, des adresses, des messages d'erreur ou des codes d'identification complexes. C'est ici qu'intervient le type Chaîne de Caractères (String).

Si le type Caractère est une brique isolée, la Chaîne est la structure qui permet de construire des édifices textuels. Mais attention : sous son apparence simple de "phrase", la chaîne de caractères cache une nature hybride.

L'objectif de ce module est de vous donner les outils pour décomposer (analyser), transformer et reconstruire n'importe quelle information textuelle, en passant de la simple lecture d'un nom à la validation complexe de formats sécurisés.

Chapitre 2. Définition et Propriétés

2.1. Définition

Une chaîne de caractères est une structure de données qui permet de stocker une suite finie de symboles (lettres, chiffres, ponctuation, espaces). Contrairement au type Caractère qui ne contient qu'un seul symbole, la chaîne peut en contenir plusieurs centaines.

Pratiquement, on peut la considérer comme un tableau de caractères particulier. Toutefois, contrairement à un tableau classique, la chaîne de caractères possède une longueur dynamique : elle peut être vide, contenir 5 caractères ou 200, dans la limite de sa capacité maximale déclarée.

Exemples

Variables

```
NomEtudiant : Chaîne  
PrenomEtudiant : Chaîne  
PhraseBienvenue : Chaîne  
CodeIdentifiant : Chaîne  
CheminFichier : Chaîne
```

Des valeurs possibles pour ces variables :

```
NomEtudiant ← "Benhmed"  
PrenomEtudiant ← "Ahmed"  
PhraseBienvenue ← "Bienvenue dans le cours d'algorithmique !"  
CodeIdentifiant ← "ALGO-2024-X12"  
CheminFichier ← "C:\Documents\Support_Cours.txt"
```

2.2. Propriétés

Propriétés clés :

- **Indice** : En Pascal, le premier caractère d'une chaîne se trouve à l'indice 1.
- **Accès** : On accède à un caractère précis avec la syntaxe MaChaine[i].
- **Type** : C'est un type "pseudo-statique". On définit souvent une taille maximale : Tableau [1..255] de Caractère ou simplement le type Chaîne[255].

2.3. Accès et Modification d'un caractère

Comme nous l'avons vu, une chaîne se comporte comme un tableau de caractères dont le premier indice est 1. On peut donc accéder à n'importe quel symbole de la chaîne en utilisant son rang (indice) entre crochets [].

2.3.1. Accès (Lecture)

Si PrenomEtudiant contient "Ahmed", alors PrenomEtudiant[1] vaut 'A' et PrenomEtudiant[3] vaut 'm'.

2.3.2. Modification (Écriture)

L'un des grands avantages est que les chaînes sont mutables : il est possible de modifier directement un caractère spécifique au sein d'une chaîne sans avoir à reconstruire toute la chaîne.

Pour modifier un caractère, on utilise l'affectation classique sur un indice précis. C'est extrêmement utile pour corriger une faute de frappe ou transformer un format.

Exemple : Correction d'une erreur

Supposons une variable contenant une date mal orthographiée :

```
MaDate ← "12/05-2024"
```

On remarque que le tiret '-' à l'indice 6 devrait être un slash '/'. Nous pouvons corriger cette erreur par :

```
MaDate[6] ← "/"
```

La nouvelle valeur de la chaîne sera :

```
MaDate vaut maintenant "12/05/2024".
```

Lors d'une modification directe, le nouveau caractère doit être du type Caractère (généralement noté entre simples quotes ' '). On ne peut pas affecter une chaîne de plusieurs caractères à un seul indice (ex: MaDate[6] ← "ABC" est impossible).

Chapitre 3. Caractères spéciaux

Au-delà des lettres et des chiffres, les chaînes de caractères accueillent des symboles dits "spéciaux". Certains sont visibles (la ponctuation, les symboles mathématiques) tandis que d'autres sont invisibles ou "de contrôle", car ils servent à structurer le texte (passer à la ligne, créer un décalage). En Pascal, ces caractères sont gérés via leur code ASCII, mais en algorithmique, nous les manipulons souvent par leur représentation directe ou par des mots-clés clairs.

Parmi les caractères spéciaux :

- L'Espace (' ') : C'est le caractère spécial le plus utilisé. Il occupe une place mémoire identique à une lettre.
- L'Apostrophe ou Guillemet simple (') : En Pascal, c'est le délimiteur de chaîne. Pour inclure une apostrophe dans une chaîne, la règle est de la doubler (ex : 'L'étudiant').
- Le Retour à la ligne (CR/LF) : Invisible à l'écran, il signale que le texte qui suit doit commencer sur une nouvelle ligne.
- La Tabulation : Utilisée pour aligner du texte ou créer des retraits de paragraphe de manière régulière.
- Les Symboles de ponctuation et opérateurs : , ; : ! ? / * + - =.
- Les Caractères de masquage : Comme l'astérisque * souvent utilisé pour l'affichage des mots de passe.

Pour les caractères que l'on ne peut pas taper (comme le retour à la ligne), on utilise souvent leur code numérique (ASCII). En Pascal, on utilise le symbole # suivi du code.

- #13 : Retour chariot.
- #10 : Saut de ligne.
- #9 : Tabulation.

Chapitre 4. Fonctions et Procédures

Standards de Manipulation

Le traitement des chaînes repose sur un ensemble de primitives intégrées. Leur maîtrise permet d'effectuer des opérations complexes de recherche et de transformation.

4.1. La Fonction Longueur (**Length**)

Cette fonction retourne un entier représentant le nombre de caractères actuellement présents dans la chaîne.

- **Syntaxe** : $R \leftarrow \text{Long}(S)$
- **Type de retour** : Entier.
- **Exemple** : Si S contient "Algorithme", $\text{Long}(S)$ retourne 10.

4.2. La Fonction de Concaténation (**Concat**)

Elle permet d'assembler deux ou plusieurs chaînes pour n'en former qu'une seule. En algorithmique, l'opérateur $+$ est souvent utilisé comme raccourci.

- **Syntaxe** : $S_Resultat \leftarrow \text{Concat}(S1, S2, \dots)$ ou $S_Resultat \leftarrow S1 + S2$
- **Type de retour** : Chaîne.
- **Exemple** : $\text{Concat}(\text{"Géo"}, \text{"graphie"})$ produit "Géographie".

4.3. La Fonction d'Extraction (**Copy**)

Cette fonction extrait une portion d'une chaîne (une sous-chaîne) sans modifier la chaîne d'origine.

- **Syntaxe** : $S_Extrait \leftarrow \text{SousChaine}(S, \text{IndiceDepart}, \text{NombreCaract})$
- **Type de retour** : Chaîne.
- **Paramètres** : IndiceDepart (position du premier caractère à prendre) et NombreCaract (longueur de la portion à extraire).
- **Exemple** : $\text{SousChaine}(\text{"Informatique"}, 3, 4)$ retourne "form".

4.4. La Fonction de Recherche (**Pos**)

Elle cherche la première occurrence d'une sous-chaîne dans une chaîne cible.

- **Syntaxe** : $P \leftarrow \text{Position}(\text{SousChaine}, \text{ChaîneCible})$
- **Type de retour** : Entier.
- **Valeur retournée** : L'indice du premier caractère de la sous-chaîne trouvée. Si la recherche échoue, la fonction retourne 0.

- **Exemple** : `Position("jour", "Bonjour")` retourne 4.

4.5. La Procédure de Suppression (**De**lete)

Contrairement aux fonctions précédentes, il s'agit d'une procédure qui modifie directement le contenu de la variable passée en paramètre.

- **Syntaxe** : `Supprimer(S, IndiceDepart, NombreCaract)`
- **Action** : Retire `NombreCaract` de la chaîne `S` à partir de `IndiceDepart`.
- **Exemple** : Si `S` vaut "Détester", l'instruction `Supprimer(S, 3, 2)` transforme `S` en "Déter".

4.6. La Procédure d'Insertion (**Insert**)

Cette procédure insère une sous-chaîne à l'intérieur d'une autre à une position donnée.

- **Syntaxe** : `Inserer(SousChaine, S_Cible, Indice)`
- **Action** : La `SousChaine` est injectée dans `S_Cible` à la position `Indice`. Le reste de la chaîne est décalé vers la droite.
- **Exemple** : Si `S` vaut "Bjr", `Inserer("onjou", S, 2)` transforme `S` en "Bonjour".

Chapitre 5. Exemples d'application

L'importance du traitement des chaînes de caractères réside dans le fait qu'elles constituent le lien principal entre l'utilisateur et la machine. La plupart des informations du monde réel (noms, adresses, codes d'identification) sont stockées sous forme de texte structuré.

Le travail de l'informaticien consiste alors en deux actions majeures : la génération (création d'un code formaté à partir de données brutes) et le parsing (analyse et décomposition d'une chaîne pour en extraire les informations utiles).

5.1. Le Matricule des Véhicules

Le matricule est une chaîne de caractères hautement structurée. Selon le format décrit (de droite à gauche) :

- Wilaya (2)
- Espace
- Année (2)
- Type (1)
- Espace
- Numéro Séquentiel.

5.1.1. Analyse (Parsing) :

À partir d'un matricule comme "12345 120 16", l'algorithme doit être capable d'extraire :

- La Wilaya (les 2 derniers caractères) : 18 (Jijel).
- L'année de mise en circulation (indices 7 et 8) : 20 (soit 2020).
- Le type de véhicule (indice 6) : 1 (Véhicule de tourisme).

5.1.2. Génération :

Lors de l'immatriculation d'un nouveau véhicule, le système assemble les composants :

- Matricule $\leftarrow$ NumSequentiel + " " + Type + Annee + " " + CodeWilaya (ou en utilisant la fonction concaténation)

5.2. Le Code-barres (EAN-13)

Le format EAN-13 est un standard mondial composé de 13 chiffres. Structure est comme suit :

- Les 3 premiers chiffres : Code pays (ex: 613 pour l'Algérie).
- Les 9 chiffres suivants : Code fabricant et code produit.
- Le dernier chiffre : Clé de contrôle (calculée mathématiquement pour vérifier l'intégrité).

Le parsing permet de vérifier instantanément la provenance d'un produit en extrayant la sous-chaîne des trois premiers caractères.

5.3. L'URL (Uniform Resource Locator)

L'URL est l'adresse d'une ressource sur le web. Son analyse est fondamentale pour les navigateurs et les serveurs web.

- Exemple : <https://www.univ-jijel.dz/ens/informatique>

5.3.1. Parsing par séparateurs :

- Tout ce qui précède `://` est le protocole (`https`).
- Tout ce qui se trouve entre `://` et le premier `/` suivant est le domaine (`www.univ-jijel.dz`).
- Le reste constitue le chemin d'accès à la ressource.

5.4. Le Relevé d'Identité Bancaire (RIB)

Le RIB est un autre exemple de format rigide indispensable aux transactions financières. Sa Structure est comme suit :

- Code Banque (3)
- Code Agence (3)
- Numéro de Compte (12)
- Clé RIB (2).

Le système de virement doit "parser" le RIB pour identifier la banque de destination et utiliser la "Clé RIB" (les deux derniers chiffres) pour valider que le numéro de compte ne contient pas d'erreur de frappe, évitant ainsi des transferts erronés.

Chapitre 6. Conclusion

La maîtrise des chaînes de caractères permet de transformer l'ordinateur d'une simple calculatrice en un système d'information capable de comprendre et d'organiser le monde réel. Cependant, manipuler des informations liées (comme les 4 parties d'un matricule ou les 4 parties d'un RIB) dans des variables séparées devient vite compliqué.

C'est ici qu'intervient la notion d'Enregistrement (Record), qui permet de regrouper ces informations sous une seule entité. Les enregistrements seront le sujet du chapitre suivant.