

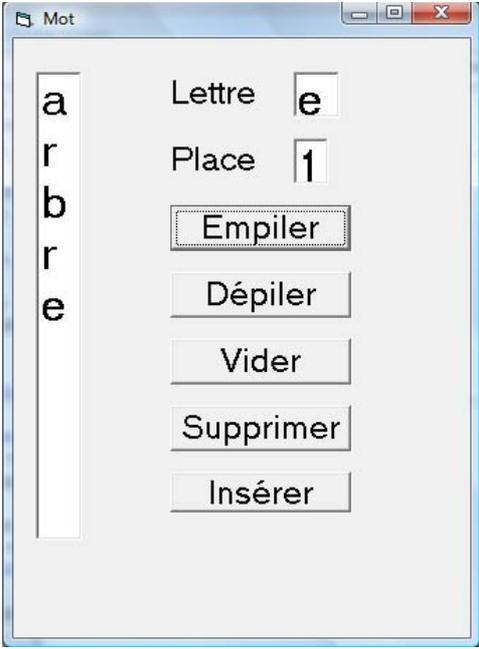
## Exercice sur les tableaux remplis partiellement

E.Thirion - 04/07/2015

L'exercice suivant est un projet à compléter dont l'interface graphique est déjà réalisée, ce qui vous permettra un gain de temps important. Ce projet est disponible par téléchargement. Pour savoir comment y accéder cliquez [ici](#).

D'autre part, ce document fait partie d'un ensemble de cours du même auteur (programmation procédurale et objet, programmation web, bases de données) auxquels vous pouvez accéder en cliquant [ici](#).

**Projet:** Exo-Tableaux/Pascal/Mot/ProjetMot.lpi

Formulaire	Nom des composants
	<p>Zone de liste :</p> <p>ZL_Mot</p> <p>Zones de texte:</p> <p>ZT_Lettre</p> <p>ZT_Place</p>

### Objectif

Il s'agit de réaliser un programme permettant de construire un mot en y ajoutant progressivement des lettres, puis de le modifier par suppression ou insertion de lettre.

Le mot est représenté par un tableau **TMot** rempli partiellement jusqu'à un indice de fin **n**. Il est déclaré comme suit:

```
Const M = 8;
var TMot : array [1..M] of String
    n : Integer;
```

**Ne modifiez pas ces déclarations** car vous devrez les utiliser tout le long de l'exercice.

L'affichage du mot dans la zone de liste est déjà réalisé. Le code se trouve dans la procédure **AfficherMot**.

Votre rôle consistera à écrire le code de chaque bouton du formulaire.

**1) Adjonction d'une lettre à la fin du mot**

Lorsque l'utilisateur clique sur le bouton **Empiler**, la lettre est ajoutée à la fin du mot.

**2) Suppression de la dernière lettre**

Lorsque l'utilisateur clique sur le bouton **Depiler**, la dernière lettre est supprimée.

**3) Effacement du mot**

Lorsque l'utilisateur clique sur le bouton **Vider**, le mot devient vide.

**4) Suppression d'une lettre (autre que la dernière)**

Lorsque l'utilisateur clique sur le bouton **Supprimer**, la lettre à la place indiquée est supprimée.

**5) Insertion d'une lettre**

Lorsque l'utilisateur clique sur le bouton **Insérer**, la lettre donnée est insérée à la place indiquée.

**6) Protection du programme contre les erreurs**

Pour terminer, protégez votre programme contre les erreurs de manipulation de l'utilisateur. Par exemple, il est interdit d'empiler plus de **M** fois, car le mot ne peut contenir que **M** lettres au maximum. Si cela se produit, on affichera un message d'erreur.

Protégez de cette manière chacun des boutons du formulaire contre les opérations qui pourrait provoquer une erreur d'exécution. Pour éviter la répétition inutile de code, il est préférable d'écrire des sous-programmes de contrôles des données. Par exemple, un sous-programme de contrôle de la lettre et un sous-programme de contrôle de la place.