

## POUR COMMENCER A QUOI VISE CE PROJET ?

Ecrire un programme en langage Pascal, qui permettra d'avoir une grille pré-remplie de fouillis (caractères), puis avoir à coté un tableau rempli des mots à rechercher dans la grille,

Aussitôt le mot trouvé, il change de couleur. A la fin on aura des caractères non-utilisés, ces caractères devront être retrouvés afin d'afficher un mot qui désignera le nom d'une ville Algérienne.

## LE FOUILLIS C'EST QUOI ?

Le fouillis est un jeu que l'on trouve dans des journaux, et qui consiste à retrouver dans une grille les mots contenus dans une liste accompagnant la grille. Les mots figurent tous dans la grille, et sont écrits horizontalement (de gauche à droite ou de droite à gauche), verticalement (de haut en bas ou de bas en haut), suivant la première diagonale ou suivant la deuxième diagonale (en montant ou en descendant). Lorsque tous les mots sont cochés, les lettres restantes constituent un mot dont la définition est donnée.

## PARTIE 1 :

Dans cette première partie du projet, on a pour tâche d'implémenter la grille de fouillis donnée dans l'exemple, ensuite de réaliser un programme qui la résout d'une manière automatique en utilisant la liste des mots qui l'accompagne.

Et pour cela nous avons eu besoin d'utiliser des variables, des constantes, une matrice qui fera office de grille,

Un tableau pour contenir tous les mots, et puis de cinq procédures.

Comme variables nous avons déclaré notre matrice (Grille :Gr) et notre tableau (TabMot :T) et une deuxième matrice (sara :S) qui servira pour l'affichage,

Et comme constantes le nombre de lignes et de colonnes dans la matrice,

Et pour éviter la répétition on a utilisé deux types qui sont : Gr= array[-1..NB\_ligne,-1..NB\_colonne] of char;

T= array[1..39] of string ;

S=array [-1..NB\_ligne,-1..NB\_colonne] of char ;

Et enfin passons aux procédures utilisées :

## PREMIERE PROCEDURE :

Cette procédure sert à remplir la grille,

Pour cela on a commencé par déclarer les données d'entrées et de sorties qui sont une grille vide en entrée et une grille pré-remplie en sortie,

Ensuite ce remplissage s'est fait grâce à 225 affectations, ou à chaque fois on affectait à une case de la grille un caractère spécifique.

## DEUXIEME PROCEDURE :

le but de cette deuxième procédure est d'afficher la grille pré-remplie ,

Et pour cela, tout comme la première procédure on a utilisé en donnée de sortie une Grille remplie. Tout cet affichage se fera grâce à deux boucles "POUR", puis pour l'afficher il suffit de l'écrire.

## TROISIEME PROCEDURE :

le but de cette dernière est de remplir un tableau : TabMot , des 39 mots à rechercher , et par la suite afficher ce tableau ,pour ceci nous avons déclaré en donnée d'entrée et de sortie un tableau : TabMot , et en variable "k" comme entier, Ensuite on a utilisé 39 affectations pour affecter à chaque ligne du tableau un mot (une chaîne de caractère), et à la fin une boucle "POUR" afin d'afficher le tableau : TabMot .

## QUATRIEME PROCEDURE :

RechMot(var Grille: Gr;var TabMot: T); en donnée d'entrée cette procédure aura une chaîne de caractères (qui sont les mots du tableau) qu'on fera défiler avec une boucle !

En premier lieu elle va parcourir la matrice pour trouver le premier caractère du mot à rechercher

Ensuite elle comparera les caractères similaires dans les 8 directions jusqu'à trouver la longueur du mot désiré , si tous les caractères correspondent c'est bon , sinon on change de direction .

{ recherche un mot dans la grille caractère par caractère dans toutes les directions. Aussitôt le mot est retrouvé il change de couleur et la grille est affichée. Cette dernière doit s'afficher à chaque fois qu'un mot est trouvé. }

## CINQUIEME PROCEDURE :

RechCar(var Grille:Gr);

à chaque fois qu'un mot est trouvé il sera affecté à la matrice d'affichage "sara",

la matrice qui servira pour l'affichage aura donc tous les mots du tableau sauf les caractères restants du dernier mot à trouver ,

donc on aura qu'à comparer les deux matrices pour trouver les caractères restants non utilisés

pour constituer le dernier mot qui est le nom d'une ville algérienne.

{ recherche les caractères restants (non utilisés) dans la grille. La recherche se fera ligne par ligne afin de retrouver le mot, dont la définition est « Une ville Algérienne », et l'afficher.} et pour cette dernière procédure on a eu besoin d'une deuxième matrice pour pouvoir afficher les caractères restants !

## Programme principal !

Maintenant il est temps de faire appel aux procédures déclarées en haut,

RempGr {remplissage de la grille}

RempTabMot {remplissage du tableau des mots à rechercher et affichage du tableau par la meme occasion}

AffichGr {affichage de la grille }

RechMot {recherche les mots}

RechCar {recherche les caractères restants}.

Le problème qui se pose dans ce projet est de réaliser la grille et le tableau qui l'accompagne, ensuite d'essayer de résoudre cette grille automatiquement grace aux procédures déclarées.

Tout ça est fait en langage PASCAL, en utilisant le logiciel MyPascal .

## PARTIE 2 :

Ensuite pour la deuxième partie du projet c'est simple, c'est plus l'interface graphique !

on nous a demandé de rendre notre programme interactif, c'est-à-dire c'est un utilisateur qui jouera à la place de l'ordinateur.

## EN CONCLUSION :

On a utilisé plein de structures de données vues en cours :les matrices, les chaines de caractères (mots),les tableaux... ; □

Et les boucles, les structures conditionnelles... ;

et meme Des bibliothèques Pascal notamment crt ;



**X**

---

OUYAHIA SARA LYNA  
Rédactrice en chef