

TP 8 Redirections - Matrices - Chaînes de caractères

Objectifs

Manipuler les entrées/sorties C (via les redirections), implémenter en C quelques algorithmes de chaînes de caractères et de matrices.

1 Un exo de matrices

Objectif : récupérer des données stockées dans un fichier dans une matrice C

1. Récupérer les deux fichiers sur la page du cours, avec `wget`.
2. Les mettre dans un répertoire TP8.
3. Compiler le fichier `getmat.c` en `getmat`
4. Ce binaire attend des informations sur le terminal (utilisation de `fgets`). On utilise donc des redirections pour donner le fichier `thegrid.txt` comme entrée :
`./getmat <thegrid.txt`.
5. Comment faire pour récupérer les sorties de votre programme dans un fichier ?

Deux fonctions à écrire rapidement

1. Écrire une fonction qui compte le nombre de fois qu'un caractère donné apparaît dans la matrice.
2. Écrire une procédure qui construit la matrice symétrique de la matrice d'entrée.
3. Tester en imprimant les résultats.

2 (Ré)écriture de texte

Préparation On fournit dans le fichier `tpchaines.c` une procédure et un programme principal :

```
#include <stdio.h>
#include <stdbool.h>

void mange_et_reecrit(void)
{
    char c;
    c = getchar();
    while ( c != EOF )
    {
        putchar(c);
    }
}
```

```

        c = getchar();
    }
}

int main(void)
{
    mange_et_reecrit();
    return 0;
}

```

On demande de compiler le programme et de l'exécuter :

1. Que fait ce programme ?
2. Écrire un fichier de nom `toto.txt` qui contient :

```

truc
bidule
plouf

```

(pas de saut de ligne après `plouf`).

3. Dans le terminal, faire `copy <toto`. Que ce passe-t-il ?
4. Que va-t-il se passer si on fait `copy <toto >titi` ? Vérifier. Comparer avec la commande `cp` d'Unix.

Remarque : on pourrait utiliser `scanf` au lieu de `getchar`.

Algorithmes de chaînes Pour les deux exercices suivants, on écrira les programmes sur le modèle précédent (**un `.c` par exercice**).

1. Écrire un programme qui compte le nombre de caractères (y compris les espaces, les sauts de lignes, les tabulations) de l'entrée standard. Utiliser les redirections pour compter le nombre de caractères du fichier `toto`.
2. Écrire un programme qui imprime les lettres qui ne sont pas apparues dans le fichier (on suppose que les lettres sont uniquement des minuscules).
Indication : On pourra utiliser un tableau de 26 cases que l'on mettra à jour à chaque fois que l'on rencontrera une lettre. Comment trouver l'indice de la case à mettre à jour ?
3. Écrire un programme qui écrit le fichier à l'envers. Sur le fichier `toto`, le programme donnera :

```

fuolp
eludib
curt

```

Indication : On se demandera comment mémoriser les caractères du fichier

4. Écrire un programme qui dit si un certain mot (un "motif") est présent dans le fichier.