

TD : feuille 10. Les Tableaux (corrigé)

Université Paris-Nord, Institut Galilée. DEUG MIA1, 2002/03

Exercice 1.

1. Écrire un programme en C permettant de saisir une suite de caractères dans un tableau. On considérera que la suite se termine lorsque l'utilisateur effectue un retour à la ligne (caractère '\n') et que la suite a une longueur maximale définie par une constante TAILLELIGNE.
2. Compléter le programme précédent afin d'afficher la suite de caractères saisie par l'utilisateur ainsi que le nombre de caractères saisis ('\n' non compris).

Solution :

```
#include <stdio.h>
#define TAILLEMAX 200
#define FAUX 0
#define VRAI 1

int main() {
    int longueur=0;
    int finligne=FAUX;
    int i; //i : indice dans phrase.
    char phrase[TAILLEMAX];

    /* saisie d'une phrase*/
    printf("Entrez une phrase.\n");
    while((longueur<TAILLEMAX) && (finligne==FAUX))
        {scanf("%c",&phrase[longueur]);
        if(phrase[longueur]=='\n')
            finligne=VRAI;
        else
            longueur=longueur+1;
        }

    /* affichage de la phrase */
    printf("Vous avez saisie les %d caracteres suivants\n",longueur);
    for(i=0;i<longueur;i++)
        printf("%c",phrase[i]);
    printf("\n");

    return 0;
}
```

Exercice 2.

Écrire un programme en C permettant de déterminer et d'afficher le nombre de fois que chaque lettre de l'alphabet est présente dans un tableau contenant des caractères. Le tableau contenant des caractères sera saisi par l'utilisateur. Par ailleurs, on ne distinguera pas les majuscules des minuscules. Les caractères n'appartenant pas à l'alphabet seront comptés à part.

Solution :

```
#include <stdio.h>
#define TAILLEMAX 200
#define NBLETTRE 27
#define FAUX 0
#define VRAI 1

int main() {
    int longueur=0;
    int i,j; //i : indice dans phrase, j: indice dans histo.
    int finligne=FAUX;
    char phrase[TAILLEMAX];
    int histo[NBLETTRE];

    /* saisie d'une phrase*/
    printf("Entrez une phrase.\n");
    while((longueur<TAILLEMAX) && (finligne==FAUX))
        {scanf("%c",&phrase[longueur]);
        if(phrase[longueur]=='\n')
            finligne=VRAI;
        else
            longueur=longueur+1;
        }

    /* histogramme des lettres */

    for(j=0;j<NBLETTRE;j++)
        histo[j]=0;

    for(i=0;i<longueur;i++)
        {if('a'<=phrase[i] && phrase[i]<='z' )
            j=phrase[i]-'a';
        else if ('A'<=phrase[i] && phrase[i]<='Z' )
            j=phrase[i]-'A';
        else
            j=NBLETTRE-1;

        histo[j]=histo[j]+1;
        }

    /* affichage de l'histogramme */
    for(j=0;j<NBLETTRE-1;j++)
        printf("%c apparait %d fois.\n",'a'+j,histo[j]);
    printf("il y a %d signes non reconnus\n",histo[NBLETTRE-1]);
}
```

```
    return 0;
}
```

Exercice 3.

Écrire un algorithme permettant d'afficher les mots contenus dans une phrase saisie par un utilisateur. Pour simplifier, on supposera que les mots sont uniquement séparés par un ou plusieurs espaces (caractère ' '). On ne tient pas compte de la ponctuation.

Solution :

```
#include <stdio.h>
#define TAILLEMAX 200
#define FAUX 0
#define VRAI 1

int main() {
    int longueur=0,longueurMot;
    int i,k; //i : indice dans phrase, k: indice dans mot.
    int finligne=FAUX;
    char phrase[TAILLEMAX];
    char mot[TAILLEMAX];

    /* saisie d'une phrase*/
    printf("Entrez une phrase.\n");
    while((longueur<TAILLEMAX) && (finligne==FAUX))
        {scanf("%c",&phrase[longueur]);
        if(phrase[longueur]=='\n')
            finligne=VRAI;
        else
            longueur=longueur+1;
        }

    /* decoupage en mots */

    printf("La phrase \n");
    for(i=0;i<longueur;i++)
        printf("%c",phrase[i]);
    printf("\n");
    printf("est composee des mots:\n");

    for(i=0,longueurMot=0;i<=longueur;i++)
        {if(phrase[i]!=' ' && phrase[i]!='\n') /*le test sur \n sert a afficher
        le dernier mot de la phrase */
            {mot[longueurMot]=phrase[i];
            longueurMot=longueurMot+1;
            }
        else
            {for(k=0;k<longueurMot;k++)
                printf("%c",mot[k]);
            if (longueurMot!=0)
                printf("\n");
            }
        }
}
```

```

        longueurMot=0;
    }
}

return 0;
}

```

Exercice 4.

1. Écrire un programme en C permettant à un utilisateur de saisir une matrice carrée. Cette matrice aura une taille maximum donnée par une constante TAILLEMAX. L'utilisateur devra dans un premier temps dire la taille de la matrice qu'il veut saisir.
2. Compléter le programme précédent afin d'afficher la matrice saisie par l'utilisateur.
3. Compléter le programme précédent afin de pouvoir déterminer le coefficient ayant le plus grand module parmi tous les coefficients présents sur une colonne choisie par l'utilisateur. Faire en sorte que le programme affiche la valeur de ce coefficient et la ligne à laquelle il appartient.
4. Compléter le programme précédent afin de permettre à l'utilisateur d'intervertir une ligne de son choix avec celle que vous avez déterminé à la question précédente. Faire en sorte que le programme affiche la matrice obtenue après cette interversion.

Solution :

```

#include <stdio.h>
#define TAILLEMAX 100

int main(){
    float matrice[TAILLEMAX][TAILLEMAX];
    float lignetmp[TAILLEMAX];
    float max;
    int taille;
    int i,j;
    int i1,imax; // lignes a intervertir

    /* saisie d'une matrice carree */

    printf("Quelle est la taille de la matrice carree?\n");
    scanf("%d",&taille);

    for(i=0;i<taille;i++) for(j=0;j<taille;j++)
        {printf("matrice(%d,%d)=",i+1,j+1);
        scanf("%f",&matrice[i][j]);
        }

    /* affichage de la matrice */
    printf("Vous avez saisie\n");
    for(i=0;i<taille;i++)
        {for(j=0;j<taille;j++)
            printf("%f \t",matrice[i][j]);
        printf("\n");
        }
}

```

```

/* determination du max */
do{
    printf("Quelle est la colonne dont vous voulez le max (en module)?\n");
    scanf("%d",&j);
}while(j<=0 || j>taille);
j=j-1;

max=matrice[0][j];
for(i=0;i<taille;i++)
    {if(max<matrice[i][j])
        {max=matrice[i][j];
        imax=i;
        }
    else if(max< -matrice[i][j])
        {max=-matrice[i][j];
        imax=i;
        }
    }

printf("Le max de la colonne %d vaut %f. Il est atteint a la ligne %d\n",j+1,max,imax+1);

/* intervertion des lignes */
do{
    printf("Quelle est la ligne a intervertir? ");
    scanf("%d",&i1);
}while(i1<=0 || i1>taille);
i1=i1-1;

for(j=0;j<taille;j++)
    lignetmp[j]=matrice[i1][j];
for(j=0;j<taille;j++)
    matrice[i1][j]=matrice[imax][j];
for(j=0;j<taille;j++)
    matrice[imax][j]=lignetmp[j];

/* affichage de la matrice */
printf("La nouvelle matrice est\n");
for(i=0;i<taille;i++)
    {for(j=0;j<taille;j++)
        printf("%f \t",matrice[i][j]);
    printf("\n");
    }

return 0;
}

```