
Licence E.E.A.

TD N°4 de Langage C/C++

Ce TD a pour objectif de vous faire découvrir et utiliser le langage C et peut être C++. Il s'agira de transcrire sous la forme d'un programme C, des algorithmes afin de réaliser les fonctions demandées.

1 Échanger deux valeurs

Contrairement à ce que vous pourriez penser au premier abord, ce programme est fondamental car il souligne l'importance des pointeurs dans le C. En effet, si on veut modifier les arguments d'une procédure dans cette même procédure, il faut passer non pas la valeur des arguments mais leur adresse, autrement dit un pointeur. C'est la seule manière de procéder en C et il vous faudra donc bien comprendre ce mécanisme. C'est d'ailleurs ce même procédé que vous utilisez dans la fonction `scanf` : `scanf("%d",&variable)` indique en fait que l'on passe l'adresse de variable pour que `scanf` modifie le contenu de variable, en récupérant un entier dans cet exemple.

Notions introduites : le pointeur, accès à un objet dont on connaît l'adresse, passage des adresses en paramètres de fonctions, le type float.

2 Ordonner

Le but de cet exercice est de lire deux réels puis d'imprimer ces deux réels, rangés par ordre décroissant. L'exemple ci-dessous donne la structure générale du programme.

```
#include<stdio.h>

/* PROTOTYPE(S)*/

int main()
{
  /*A COMPLETER */
}

void ordonne (float *x, float *y)
{
  /*A COMPLETER */
}
```

3 Renverser une chaîne de caractères

Le but de cet exercice est de "renverser" un tableau de caractères, considéré comme une chaîne de caractères et donc terminé par `\0`. Il est imposé, dans cet exercice, d'utiliser la fonction "echanger", prototypée et définie dans l'exemple en l'adaptant à l'échange de caractères.

Exemple :

```
/* Il s'agit de saisir dans le tableau "mot" un mot d'au plus 20 lettres, de
l'inverser et d'afficher le mot inversé.
Il est imposé d'utiliser une fonction qui fait l'échange de deux caractères.*/
```

```

#include <stdio.h>

enum {LONGMAX=21}; /*Cela permettra d'avoir des mots de 20 lettres au plus.
    Il faut garder dans le tableau qui contiendra le mot,
    une case pour le caractère '\0' qui doit terminer toute
    chaîne de caractères*/

void echanger(char *, char *);
void inverser(void);

char mot[LONGMAX];

void main()
{
    printf("Donnez le mot à inverser (au plus %d caractères) : ",LONGMAX-1);
    scanf("%s",mot); /*lit une chaîne de caractères et la termine par '\0'*/
    inverser();
    printf("Après inversion, on a : %s\n",mot);
}

void inverser()
{
    /*A COMPLETER*/
}

void echanger(char * ad_c1, char * ad_c2)
{
    /*A COMPLETER*/
}

```

4 Concaténer deux chaînes de caractères

Le but de ce programme est de "concaténer" deux chaînes de caractères, représentant par exemple prénom et nom. Pour stocker les données, de type chaînes de caractères, une allocation de mémoire statique est prévue. Le nombre maximum de caractères de chaque chaîne est fixé à 10. Pour le résultat, on veut seulement le voir imprimé : la place qu'il occupe est attribuée dynamiquement puis libérée.

Notions introduites : allocation dynamique de la mémoire, libération de cette mémoire, passage des tableaux en paramètres de fonctions, manipulation des chaînes de caractères.

5 Tri sans variable globale

Le but de cet exercice est de reprendre le programme de tri de l'exemple 3 pour qu'il ne contienne plus de variable globale.

```

/*Vous trouverez ci-dessous le programme de tri de l'exemple 3.
Il s'agit de le transformer pour qu'il n'ait plus aucune variable globale.*/

#include <stdio.h>

enum {MAXDONNEES=8};

int tab[MAXDONNEES];

int lire(void);

```

```
void trier(int);
void afficher(int);

void main()
{
    int NbrDonnees;

    NbrDonnees=lire();
    trier(NbrDonnees);
    afficher(NbrDonnees);
}

int lire()
{
    int donnee,nbr=0;

    printf("Entrez les donnees entieres positives a trier, sur une ligne, "
           "separees par des blancs.\nTerminez la saisie en tapant -1.\n"
           "Vous avez droit a au plus %d valeurs\n",MAXDONNEES);
    scanf("%d",&donnee);
    while((nbr<=MAXDONNEES) &&(donnee!=-1))
    {
        if (nbr==MAXDONNEES)
        {
            printf("\nvous avez donne plus de %d entiers, seuls les %d "
                   "premiers seront tries\n",MAXDONNEES,MAXDONNEES);
            return nbr;
        }
        else
        {
            tab[nbr]=donnee;
            nbr++;
            scanf("%d",&donnee);
        }
    }
    return nbr;
}

void trier(int nombre)
{
    int i,j,cle;

    for (i=1;i<nombre;i++)
    {
        cle=tab[i];
        for (j=i-1;(j>=0) && (cle<tab[j]);j--) tab[j+1]=tab[j];
        tab[j+1]=cle;
    }
}

void afficher(int nombre)
{
    int i;

    printf("\nvoici le tableau trie\n");
    for(i=0;i<nombre;i++) printf("%5d",tab[i]);
    printf("\n");
}
```

6 Concaténer avec des tableaux dynamiques

Le but de cet exercice est de concaténer deux chaînes de caractères; ces deux chaînes de caractères ont été stockées dans des tableaux alloués dynamiquement. Une fonction est chargée de la concaténation des deux chaînes. Si la taille du premier tableau est suffisante pour stocker les deux chaînes de caractères, c'est ce qui est fait; sinon, on effectue une nouvelle allocation mémoire pour y mettre la concaténation des deux chaînes; dans les deux cas, les tableaux autres que celui contenant la concaténation des deux chaînes seront désalloués.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

const int ESTIMATION=15; /*nombre maximum estimé de caractères pour le prénom
                           comme pour le nom*/
/*PROTOTYPE(S)*/

void main()
{
    char * prenom, * nom;
    char * prenom_nom;

    /*ALLOCATION DYNAMIQUE POUR PRENOM*/

    printf("Donnez le prénom (au plus %d caractères) : ",ESTIMATION);
    fgets(prenom,ESTIMATION+1,stdin);
    if (prenom[strlen(prenom)-1]=='\n') prenom[strlen(prenom)-1]='\0';
    fseek(stdin,0,2);

    printf("Donnez le nom (au plus %d caractères) : ",ESTIMATION);

    /*INSTRUCTIONS SIMILAIRES POUR LE NOM*/

    puts(prenom_nom);
    free(prenom_nom);
}

char * AlloueEtConcat(char * s1, char * s2)
{
    int longueur;
    char * s3;

    /*A COMPLETER, SANS OUBLIER DE LIBERER AUTANT DE MEMOIRE QUE POSSIBLE*/
}
```