Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation en langage C

Travaux Dirigés & Solutions

Professeur BALOUKI Youssef
Département Mathématiques et Informatique

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation
TD 1

Exercice 1
Ecrire un algorithme qui détermine si une variable entière est paire ou impaire.

Exercice 2
Ecrire un algorithme qui saisit deux variables entières, échange leurs contenus et les affiche avant et après l'échange.

Exercice 3
Ecrire un algorithme qui demande trois nombres réels à l'utilisateur et les affiche dans un ordre croissant.

Exercice 4
Écrire un algorithme qui, après avoir permis la saisie de 3 variables réelles X, Y et Z, effectue une permutation circulaire de leurs valeurs respectives (i.e. Z = Y, X = Z, Y = X).

Exercice 5
Ecrire l'algorithme qui lit un entier positif inférieur à 999 (composé de trois chiffres au maximum) et d'afficher le nombre de centaines, de dizaines et d'unités.

Exercice 6
Ecrire un algorithme qui demande successivement 5 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand et le plus petit parmi ces 5 nombres.

Exercice 7
Ecrire l'algorithme qui permet de saisir la moyenne générale d'un étudiant et de déterminer son résultat et sa mention.

Exercice 8
Ecrire un algorithme qui teste si une année est bissextile ou non.

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation TD2

Exercice 1

Ecrivez un programme qui lit N nombres entiers au clavier et qui affiche leur somme, leur produit et leur moyenne. Choisissez un type approprié pour les valeurs à afficher. Le nombre N est à entrer au clavier.
Résolvez ce problème,
a) en utilisant while,
b) en utilisant do - while,
c) en utilisant for.
d) Laquelle des trois variantes est la plus naturelle pour ce problème?
e) Répétez l'introduction du nombre N jusqu'à ce que N ait une valeur entre 1 et 15.
Quelle structure répétitive utilisez-vous? Pourquoi?

Exercice 2

Calculez par des soustractions successives le quotient entier et le reste de la division entière de deux entiers entrés au clavier.

Exercice 3

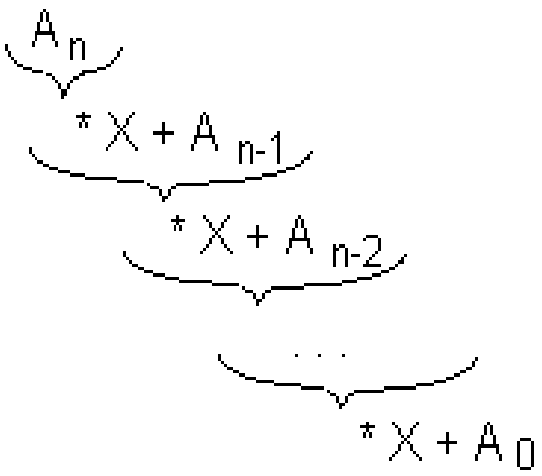
Calculez la factorielle N! = 123...(N-1)N d'un entier naturel N en respectant que 0!=1.
a) Utilisez while,
b) Utilisez for.

Exercice 4

Calculez par multiplications successives X^N de deux entiers naturels X et N entrés au clavier.

Exercice 5

Calculez pour une valeur X donnée du type float la valeur numérique d'un polynôme de degré n:
P(X) = AnXn + An-1Xn-1 + ... + A1X + A0
Les valeurs de n, des coefficients An, ... , A0 et de X seront entrées au clavier.
Utilisez le schéma de Horner qui évite les opérations d'exponentiation lors du calcul:



Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Programmation en langage C TD3

Exercice 1

Ecrire un programme qui lit une ligne de texte (ne dépassant pas 150 caractères) la mémorise dans une variable TXT et affiche ensuite:

1. la longueur L de la chaîne.
2. le nombre de 'e' contenus dans le texte.
3. toute la phrase à rebours, sans changer le contenu de la variable TXT.
4. toute la phrase à rebours, après avoir inversé l'ordre des caractères dans TXT:
   a. voici une phrase !
   b. ! esarhp enu iciov

Exercice 2

Ecrire un programme qui lit un texte TXT (de moins de 150 caractères) et qui enlève toutes les apparitions du caractère 'm' en tassant les éléments restants. Les modifications se feront dans la même variable TXT.

Exemple: Cette ligne contient quelques lettres e.
Ctt lign contint qulqus lttrs.

Exercice 3

Ecrire un programme qui lit 10 mots et les mémorise dans un tableau de chaînes de caractères. Trier les 10 mots lexicographiquement en utilisant les fonctions strcmp et strcpy. Afficher le tableau trié.

Exercice 4

Ecrire un programme qui lit deux tableaux A et B et leurs dimensions N et M au clavier et qui ajoute les éléments de B à la fin de A. Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exercice 5

Ecrire un programme qui lit une chaîne de caractères CH et détermine la longueur de la chaîne à l'aide d'un pointeur P. Le programme n'utilisera pas de variables numériques.

Exercice 6

Ecrire de deux façons différentes, un programme qui vérifie sans utiliser une fonction de <string>, si une chaîne CH introduite au clavier est un palindrome:
a) en utilisant uniquement le formalisme tableau
b) en utilisant des pointeurs au lieu des indices numériques

Exercice 7

Ecrire un programme qui lit une chaîne de caractères CH au clavier et qui compte les occurrences des lettres de l'alphabet en ne distinguant pas les majuscules et les minuscules. Afficher seulement le nombre des lettres qui apparaissent au moins une fois dans le texte.
Exemple:

```
Entrez une ligne de texte (max. 100 caractères) :
Papa
La chaîne "Papa" contient :
2  fois la lettre 'A'
2   fois la lettre 'P'
```

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation TD 4

Exercice 1

Soit F la fonction numérique définie par $F(X) = X^3+1$. On désire construire un tableau de valeurs de cette fonction. Le nombre N de valeurs ainsi que les valeurs de X sont entrés au clavier par l'utilisateur.
En modularisant ce problème, nous obtenons un programme principal très court et bien 'lisible'. La fonction main joue le rôle du programme principal:

```
main()
{
 float X[100];   /* valeurs de X */
 float V[100];   /* valeurs de F(X) */
 int N;
 LIRE_DIM(&N);   /* 1 <= N <= 100 */
 LIRE_VECTEUR(X, N);
 CALCULER_VALEURS(X, V, N);
 AFFICHER_TABLE(X, V, N);
 return 0;
}
```

Définir les quatre fonctions suivantes: LIRE_DIM, LIRE_VECTEUR, CALCULER_VALEURS et AFFICHER_TABLE.

Exercice 2 :

Ecrire une fonction permettant de remplir les fiches de tous les étudiants de la classe.

Exercice 3

Ecrire la fonction **NMOTS** qui retourne comme résultat le nombre de mots contenus dans une chaîne de caractères CH. Utiliser une variable logique, la fonction **isspace** et une variable d'aide N.

Exercice 4

Ecrire la fonction **INSERER** qui place un élément X à l'intérieur d'un tableau qui contient N éléments triés par ordre croissant, de façon à obtenir un tableau à N+1 éléments triés par ordre croissant. La dimension du tableau est incrémentée dans la fonction INSERER.

Ecrire un programme pour tester la fonction INSERER.

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation TD5

Exercice 1

Ecrire un programme qui détermine dans un texte dont le nom est entré au clavier:

- ✓ le nombre de caractères qu'il contient,
- ✓ le nombre de chacune des lettres de l'alphabet (sans distinguer les majuscules et les minuscules),
- ✓ le nombre de mots,

Les retours à la ligne ne devront pas être comptabilisés dans les caractères. On admettra que deux mots sont toujours séparés par un ou plusieurs des caractères suivants:

- fin de ligne
- espace
- ponctuation: . : , ; ? !
- parenthèses: ( )
- guillemets: "
- apostrophe: '

Utiliser une fonction d'aide F_SEP qui décide si un caractère transmis comme paramètre est l'un des séparateurs mentionnés ci-dessus. F_SEPA restituera la valeur 1 si le caractère est un séparateur et 0 dans le cas contraire. F_SEPA utilise un tableau qui contient les séparateurs à détecter.

Exercice 2

Soit le type structure suivant, représentant un point d'un plan :

```
struct point
        { char c ; // nom attribué au point
          int x, y ; // ses coordonnées
         }
```

Écrire une fonction qui reçoit en argument l'adresse d'une structure du type point et qui renvoie en résultat une structure de même type correspondant à un point de même nom et de coordonnées opposées.

Écrire un petit programme d'essai.

Problème 1

Soit un fichier texte contenant des données organisées dans des enregistrements de longueur fixe.

	Code client	Nom	Age
Enrg1	12	LATIF	23
Enrg2	13	BELHAJ	56
Enrg3	23	FILALI	23

Ecrire un programme qui permet de gérer le contenu du fichier **client.txt** grâce au menu suivant :

1 : Ajouter un nouveau client
2 : Lister tous les clients
3 : Chercher un client
4 : Modifier le nom d'un client
5 : Supprimer un client
6 : Quitter

Problème 2

Ecrire un programme qui vérifie la validité d'une série de numéros de comptes mémorisés dans un fichier(compte.txt). Un numéro de compte composé de trois parties: un numéro, un séparateur '-' et un numéro de contrôle. Un numéro de compte est correct:

- si le reste de la division entière de la valeur devant le séparateur '-' par 97 est différent de zéro et égal à la valeur de contrôle.

- si le reste de la division par 97 est zéro et la valeur de contrôle est 97.

Exemple:

Nombre de CCP 15742-28 : 15742 modulo 97 = 28 correct

Nombre de CCP 72270-5 : 72270 modulo 97 = 5 correct

Nombre de CCP 22610-10 : 22610 modulo 97 = 9 incorrect

Nombre de CCP 50537-0 : 50537 modulo 97 = 0
nombre incorrect, car la valeur de contrôle devrait être 97.
Nombre de CCP 50537-97 : 50537 modulo 97 = 0 correct

Utiliser une fonction Compte_TEST qui obtient comme paramètres les deux parties numériques d'un nombre de CCP et qui affiche alors un message indiquant si le numéro de compte est valide ou non.

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation en langage C

Solution des Travaux Dirigés

Professeur BALOUKI Youssef
Département Mathématiques et Informatique

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Algorithmique et Programmation

Solution TD 1

Exercice 1

```
var :
nombre,n : Entier
Debut
   ecrire "un nombre"
   lire nombre
     Si (x mod 2 = 0) Alors
       Ecrire(x, " est pair");
   Sinon
       Ecrire(x, " est impair");
   FinSi
Fin
```

Exercice 2

Échange de deux variables entières avec 1 variable temporaire

```
Var :
variable_1, variable_2 : entier
ancien_1 : entier
{
// Saisie des valeurs
Afficher(“Saisir la valeur de la première variable : “)
Saisir(variable_1)
Afficher(“Saisir la valeur de la deuxième variable : “)
Saisir(variable_2)
// Permutation
ancien_1 <- variable_1
variable_1 <- variable_2
variable_2 <- ancien_1
// Affichage du résultat
Afficher(“La valeur de la première variable est maintenant : “, variable_1)
Afficher(“La valeur de la deuxième variable est maintenant : “, variable_2)
}
```

Exercice 3

```
Si a < b
   Si b < c
      Afficher a b c
   Sinon
      Si a < c
         Afficher a c b
      Sinon
         Afficher c a b
Sinon
   Si c < b
      Afficher c b a
   Sinon
      Si a < c
         Afficher b a c
      Sinon
```

Afficher b c a

Exercice 4

Algorithme permutation 3 nombres
{permute 3 nombres}
Variables X, Y, Z, Temporaire: Entier
Début
Afficher(« saisir X : ») ;
Saisir(X) ;
Afficher(« saisir Y : ») ;
Saisir(Y) ;
Afficher(« saisir Z : ») ;
Saisir(Z) ;
Afficher(« Avant permutation »)
Afficher(« X vaut : »,X, « Y vaut : »,Y, « Z vaut : »,Z)
Temporaire ← Z
Z ← Y
Y ← X
X ← Temporaire
Afficher(« Après permutation ») ;
Afficher(« X vaut : »,X, « Y vaut : »,Y, « Z vaut : »,Z)
Fin

Exercice 5

Algorithme affichage (nombre de centaines, de dizaines et d'unités)

Variables X, Y, Z, Temporaire: **Entier**
Début
Afficher(« nombre positif inférieur à 999 : ») ;
Saisir(X) ;
Afficher(« Les unités : », X mod 10) ;

Y← X/10
Afficher(« Les dizaines : », Y mod 10) ;
Afficher(« Les centaines : », Y / 10) ;
Fin

Exercice 6

var :
nombre :Entier
max ,min : Entier
Afficher(“Saisir un nombre : “)
Saisir(nombre)
max ←nombre
mmin ←nombre

```
pour i variant de 1 à 4
```
Afficher(“Saisir un nombre : “)
Saisir(nombre)

```
si max < nombre alors Max <- nombre fin si
si min > nombre alors min <- nombre fin si
```

```
fin
```

Exercice 7

Exercice 8

```
Algorithme  annne_bissextile;
Var
     annee :réels;
Debut
            Ecrire('entrer  l'année : ') ;
            Lire(annee) ;
   Si  ((annee mod 4 =0 et annee mod 100 <> 0) ou annee mod 400 =0 )  alors
      Ecrire('l'année que vous avez entrer est bissextile .');
   Sinon
       Ecrire('l'année que vous avez entrer n' est pas  bissextile .');
   Finsi
fin
```

Algorithmique et Programmation

Solution TD numéro 2

Exercice 1

a) en utilisant **while**,

```
#include <stdio.h>
main()
{
 int N;        /* nombre de données */
 int NOMB;     /* nombre courant    */
 int I;        /* compteur */
 long SOM;     /* la somme   des nombres entrés */
 double PROD; /* le produit des nombres entrés */

 printf("Nombre de données : ");
 scanf("%d", &N);

 SOM=0;
 PROD=1;
 I=1;
 while(I<=N)
    {
     printf("%d. nombre : ", I);
     scanf("%d", &NOMB);
     SOM  += NOMB;
     PROD *= NOMB;
     I++;
    }

 printf("La somme   des %d nombres est %ld \n", N, SOM);
 printf("Le produit des %d nombres est %.0f\n", N, PROD);
 printf("La moyenne des %d nombres est %.4f\n", N, (float)SOM/N);
 return 0;
}
```

b) en utilisant **do - while**,

Remplacez le bloc de traitement (en gras) de (a) par :

```
 SOM=0;
 PROD=1;
 I=1;
 do
    {
     printf("%d. nombre : ", I);
     scanf("%d", &NOMB);
     SOM  += NOMB;
     PROD *= NOMB;
     I++;
    }
```

```
 while(I<=N);
```

c) en utilisant **for**.

Remplacez le bloc de traitement (en gras) de (a) par :

```
 for (SOM=0, PROD=1, I=1 ; I<=N ; I++)
    {
     printf("%d. nombre : ", I);
     scanf("%d", &NOMB);
     SOM  += NOMB;
     PROD *= NOMB;
    }
```

d) Laquelle des trois variantes est la plus naturelle pour ce problème?

La structure **for** est la plus compacte et celle qui exprime le mieux l'idée de l'algorithme. D'autre part, elle permet d'intégrer très confortablement l'initialisation et l'incrémentation des variables dans la structure.

Exercice 2

```
#include <stdio.h>
main()
{
 int NUM;   /* numérateur   de la division entière */
 int DEN;   /* dénominateur de la division entière */
 int DIV;   /* résultat de la division entière */
 int RES;   /* reste    de la division entière */

 printf("Introduisez le numérateur   : ");
 scanf("%d", &NUM);
 printf("Introduisez le dénominateur : ");
 scanf("%d", &DEN);

 RES=NUM;
 DIV=0;
 while(RES>=DEN)
    {
     RES-=DEN;
     DIV++;
    }

/* ou mieux encore : */
/*
 for (RES=NUM, DIV=0 ; RES>=DEN ; DIV++)
      RES-=DEN;
 */

 printf(" %d divisé par %d est %d reste %d\n", NUM, DEN, DIV, RES);
 return 0;
}
```

Exercice 3

Solution combinée :

(Essayez l'une ou l'autre des solutions en déplaçant les marques des commentaires !)

```
#include <stdio.h>
main()
{
 int  N;       /* La donnée   */
 int  I;       /* Le compteur */
 double FACT; /* La factorielle N! - Type double à */
              /* cause de la grandeur du résultat. */

 do
    {
      printf("Entrez un entier naturel : ");
     scanf("%d", &N);
    }
 while (N<0);

 /* a */
 /* Pour N=0, le résultat sera automatiquement 0!=1 */
 I=1;
 FACT=1;
 while (I<=N)
       {
        FACT*=I;
        I++;
       }

 /* b */
 /* Pour N=0, le résultat sera automatiquement 0!=1 */
 /*
 for (FACT=1.0, I=1 ; I<=N ; I++)
        FACT*=I;
  */

 printf ("%d! = %.0f\n", N, FACT);
 return 0;
}
```

Exercice 4

```
#include <stdio.h>
main()
{
 int  X, N;    /* Les données */
 int  I;       /* Le compteur */
 double RESU; /* Type double à cause de la */
              /* grandeur du résultat.     */

 do
    {
      printf("Entrez l'entier naturel X : ");
     scanf("%d", &X);
    }
 while (X<0);
 do
    {
      printf("Entrez l'exposant       N : ");
     scanf("%d", &N);
    }
 while (N<0);
```

```
 /* Pour N=0, le résultat sera automatiquement X^0=1 */
 for (RESU=1.0, I=1 ; I<=N ; I++)
        RESU*=X;

 /* Attention: Pour X=0 et N=0 , 0^0 n'est pas défini */
 if (N==0 && X==0)
     printf("zéro exposant zéro n'est pas défini !\n");
 else
     printf("Résultat : %d ^ %d = %.0f\n", X, N, RESU);
  return 0;
}
```

Exercice 5

```
#include <stdio.h>
main()
{
 int N;      /* degré du polynôme */
 float X;    /* argument          */
 float A;    /* coefficients successifs du polynôme */
 float P;    /* coefficient courant du terme Horner */

 printf("Entrer le degré  N du polynôme   : ");
 scanf("%d", &N);
 printf("Entrer la valeur X de l'argument : ");
 scanf("%f", &X);

 for(P=0.0 ; N>=0 ; N--)
   {
     printf("Entrer le coefficient A%d : ", N);
    scanf("%f", &A);
    P = P*X + A;
   }

 printf("Valeur du polynôme pour X = %.2f : %.2f\n", X, P);
 return 0;
}
```

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Solution TD numéro 3

Exercice 1

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//############### Exercice 1 TD 6
main() {
 char Tab[150];
 char *p;
 int i=0,nbr;
 int longueur=0;
 printf("Donner votre chaine!!\n");
 gets(Tab);
                                        // La question 1
 //############ la longueur de la chaine
 while(Tab[longueur])longueur++;
 printf("la longueur de votre chaine est : %d\n",longueur);
 getch();
 //############  la longueur en utilisant  la notion du pointeur
 p=Tab;
 while(*p)p++;
 longueur = p-Tab;
 printf("la longueur de votre chaine est : %d\n",longueur );

                                        // La question 2
//############  le nombre de 'e' contenu dans le texte
 nbr=0;
 for(i=0;i<longueur;i++)  {   if(Tab[i]=='e') nbr++;  }
 printf("le nombre de 'e' contenus dans le texte : %d\n",nbr );

 //############  le nombre de 'e' contenu dans le texte en utilisant le pointeur
 nbr=0;
 for(p=Tab;*p;p++)  {   if(*p=='e') nbr++;  }
 printf("le nombre de 'e' contenus dans le texte : %d\n",nbr );

                                        // la question 3
 //########  la phrase à rebours, sans changer le contenu de la variable Tab.
 for(i=longueur-1;i>=0;i--) printf("%c",Tab[i]) ;
 printf("\n") ;  // retour à la ligne
 //########  la phrase à rebours, sans changer le contenu de la variable Tab.
 // en utlisant le pointeur
 for(p=Tab+longueur-1;p>=Tab;p--) printf("%c",*p) ;
                                        // la question 4
//toute la phrase à rebours, après avoir inversé l'ordre des caractères dans Tab:
 int j;
 char temp;
for(i=longueur-1,j=0;i>j;i--,j++){
                        temp =Tab[j];
```

```
                                Tab[j] = Tab[i] ;
                                Tab[i]=temp;
                                }
    printf("\n") ;  // retour à la ligne
    printf("la phrase inversée %s  :\n",Tab) ;

//toute la phrase à rebours, après avoir inversé (avec pointeur)
char *q;
for(p=Tab,q=Tab+longueur-1,j=0;q>q;q--,p++){
                                temp =*p;
                                *p = *q ;
                                *q=temp;
                                }
    printf("\n") ;  // retour à la ligne
    printf("la phrase inversée en utilisant le pointeur) %s  :\n",Tab) ;
getch();
}
```

Exercice 2

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 2 TD 2
main()
{
 char Tab[150];
 char *p,*q;
 int i=0,j=0;
 printf("Donner votre chaine!!\n");
 gets(Tab);
 for (i=0;Tab[i];i++){
     Tab[j] = Tab[i];
     if(Tab[i]!='m') j++;
     }
  Tab[j]='\0';  \\ fin de la chaine
 printf("votre chaine sans 'm' : %s\n",Tab);
 // en utilisant les pointeurs
 for (p=Tab,q=Tab;*p;p++){
      *q = *p;
     if(*p!='m') q++;
     }
  *q='\0';  \\ fin de la chaine
 printf("votre chaine sans 'm' en utilsant les pointeurs: %s\n",Tab);

getch();
}
```

Exercice 3

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 3 TD 6
main()
```

```
{
 char Tab[150];
 char  T[10][10] ={"j","i","h","a","x","e","d","c","b","a"};
 int i=0,j=0,;
 char temp[10];
 for(i=0;i<10;i++) printf(" %s",T[i]);
 //printf("Donner votre chaine!!\n");
 //gets(Tab);
 printf(" %\n");
 for (i=0;i<10;i++){
    for(j=0;j<9-i;j++){
        if(strcmp(T[j],T[j+1])>=0){
                strcpy(temp,T[j]);
                strcpy(T[j],T[j+1]);
                strcpy(T[j+1], temp);
               // printf(" T[%d] %s T[%d] %s \n",j,T[j],j+1,T[j+1]);
                }

    }

}
 for(i=0;i<10;i++) printf(" %s",T[i]);



getch();
}
```

----------Solution avec pointeur

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//############### Exercice 3 TD 6
main()
{
 char Tab[150];
 char *p,*q;
 char  *T[10] ={"z","y","x","v","u","t","j","i","l","a"};
 int i=0,j=0,;
 char *temp;
 for(i=0;i<10;i++) printf(" %s",T[i]);
 //printf("Donner votre chaine!!\n");
 //gets(Tab);
 printf(" %\n");
 for (i=0;i<10;i++){
    for(j=0;j<9-i;j++){
        if(strcmp(T[j],T[j+1])>=0){
                temp = T[j];
                T[j] = T[j+1];
                T[j+1]  = temp;
                printf(" T[%d] %s T[%d] %s \n",j,T[j],j+1,T[j+1]);
                }
```

```
    }
}
 for(i=0;i<10;i++) printf(" %s",T[i]);
getch();
}
```

Exercice 4

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 3 TD 6
main()
{

 char *p,*q;
 char  A[20] ,B[10];
 int i=0,j=0;

 printf("Donner deux  chaines!!\n");
 gets(A);
 gets(B);
 p=A;
 q=B;
 while(*p)p++;
 for(q=B;*q;q++){ *p=*q; p++;}
 *p='\0';
 printf(" %s   ",A);

getch();
}
```

Exercice 5

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 3 TD 6
main()
{

 char *p,*q;
 char  A[120] ;
 printf("Saisir un texte (120 char) !!\n");
 gets(A);
 p=A;
 while(*p)p++;

 printf(" la longeur de votre texte est %d   ",p-A);


getch();
}
```

Exercice 6

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 3 TD 6
main()
{

 char *p,*q;
 char  A[120];
 bool pal=true;
 printf("Saisir un mots (120 char) !!\n");
 gets(A);
 q=A;
 while(*q)q++;q--;
 for (p=A;p<q ;p++,q--){

 if(*p!=*q){ pal =false;break;}
}

 if(pal)  printf("c'est un palindrome ");
 else     printf("ce n'est pas un palindrome ");


getch();
}
```

Exercice 7

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
//################ Exercice 3 TD 6
main()
{

 char *p,*q;
 char  A[200];
 int   B[26];
 int i =0;

 printf("Saisir un texte (200 char) !!\n");
 gets(A);

 while(i<26){B[i]=0;i++;}
 for (p=A; *p;p++){
 if( *p >= 'A' && *p <= 'Z')  B[*p-'A']++;
 if( *p >= 'a' && *p <= 'z')  B[*p-'a']++;
}

printf("La chaine %s contient\n ",A);
```

```
for(int i=0;i<26;i++){
  if(B[i]!=0) {
  printf("%d fois la lettre %c\n",B[i],'A'+i);
   }
}
getch();
}
```

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Solution TD numéro 4

Exercice 1

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
 void LIRE_DIM(int *N);      //prototype de la fonction LIRE_DIM
 void LIRE_VECTEUR(float X[],int N);
 void CALCULER_VALEURS(float X[], float V[], int N);
 void AFFICHER_TABLE(float X[], float V[], int N);

main()
{
 float X[100];   /* valeurs de X */
 float V[100];   /* valeurs de F(X) */
 int N;
 LIRE_DIM(&N);   /* 1 <= N <= 100 */
 LIRE_VECTEUR(X, N);
 CALCULER_VALEURS(X, V, N);
 AFFICHER_TABLE(X, V, N);
 getch();
}
void LIRE_DIM(int *N){
    do
    {
       printf("Donner la dimension entre 1 et 100 !\n");
       scanf("%d",N);
    }while( *N <1 || *N>100);
 }
 void LIRE_VECTEUR(float X[],int N){
    for(int i=0;i<N;i++)
    {
       printf("Donner X[%d] :\n",i);
       scanf("%f",&X[i]);
    }
 }
 void CALCULER_VALEURS(float X[], float V[], int N){
    for(int i=0;i<N;i++)
    {
       V[i] =X[i]*X[i]*X[i]+1;
    }
 }
 void AFFICHER_TABLE(float X[], float V[], int N){
    for(int i=0;i<N;i++)
    {
      printf("F(%.2f) = %.2f \n",X[i], V[i]);
    }
 }
```

Exercice 2

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

```
#include <conio.h>
#include <string.h>
struct etudiant{
        char nom[50];
        char prenom[50];
        int age;
        };

 void remplir_fiche(struct etudiant et[]);      //prototype de la fonction remplir fiche etudiant
 void afficher_fiches(struct etudiant et[]);

main()
{
 struct etudiant et[5];
 remplir_fiche(et);
 afficher_fiches(et);
 getch();
}

 void remplir_fiche(struct etudiant et[]){
    for(int i=0;i<5;i++)
    {
       printf("Donner le nom prénom et age de l'étudiant: %d \n",i+1);
       scanf("%s%s%d",&et[i].nom,&et[i].prenom,&et[i].age);

    }
 }
 void afficher_fiches(struct etudiant et[]){
    for(int i=0;i<5;i++)
    {
       printf("l'étudiant %d : %s  %s  %d\n",i+1,et[i].nom,et[i].prenom,et[i].age);
    }
 }
```

Exercice 3

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
 main()  {
 /* Déclarations */
 char CH[101];                  /* chaîne donnée */
 char *P;                  /* pointeur d'aide */
 int N;                    /* nombre des mots */
 int DANS_MOT;       /* indicateur logique:  vrai si P pointe à l'intérieur un mot */
         /* Saisie des données */
 printf("Entrez une ligne de texte (max.100 caractères) :\n");
 gets(CH);
 /* Compter les mots */
 N=0;
 DANS_MOT=0;
 for (P=CH; *P; P++)
    if (isspace(*P))
       DANS_MOT=0;
```

```
    else if (!DANS_MOT)
        {
         DANS_MOT=1;
         N++;
        }
  /* Affichage du résultat (pour perfectionnistes) */
 printf("La chaîne \"%s\" \nest formée de %d mot%c.\n",
                                CH, N, (N==1)?' ':'s');
 getch();
}
```

Exercice 4

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
#include <string.h>
void inserer(char T[],char X,int dim);
void afficher(char T[]);

 main()  {
 /* Déclarations */
 char CH[101];                    /* chaîne donnée */
 int lg;
 printf("Saisir un texte (100 char) !!\n");
 gets(CH);
 lg = strlen(CH);
 inserer(CH,'m',lg);
 afficher(CH);
 getch();
}

void inserer(char T[],char X,int dim){
    int i=dim-1;
    while(T[i]>X) {T[i+1] =T[i];i--;}
    T[i+1]=X;
    T[dim+1]='\0';
    }
void afficher(char T[]){
    puts(T);
    }
```

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Solution TD numéro 5

Exercice 1

Ecrire un programme qui détermine dans un texte dont le nom est entré au clavier:

- ✓ le nombre de caractères qu'il contient,
- ✓ le nombre de chacune des lettres de l'alphabet (sans distinguer les majuscules et les minuscules),
- ✓ le nombre de mots,

Les retours à la ligne ne devront pas être comptabilisés dans les caractères. On admettra que deux mots sont toujours séparés par un ou plusieurs des caractères suivants:

- fin de ligne
- espace
- ponctuation: . : , ; ? !
- parenthèses: ( )
- guillemets: "
- apostrophe: '

Utiliser une fonction d'aide F_SEP qui décide si un caractère transmis comme paramètre est l'un des séparateurs mentionnés ci-dessus. F_SEPA restituera la valeur 1 si le caractère est un séparateur et 0 dans le cas contraire. F_SEPA utilise un tableau qui contient les séparateurs à détecter.

Solution 1 :

```
#include <stdio.h>
main()
{

  int F_SEPA(char C);   /* Prototype de la fonction FIN_PHRASE */
   /* Déclarations : */

  char NOM_FICH[30];
  FILE *PF;

  char C;       /* caractère lu dans le fichier */
  int ABC[26];  /* compteurs des lettres de l'alphabet */
   int NTOT;     /* nombre total des caractères */
  int NAUTRES;  /* nombre des caractères qui ne font pas
                   partie de l'alphabet */
  int NMOTS;    /* nombre des mots */
  int I;        /* indice d'aide */
  int DANS_MOT; /* vrai si la tête de lecture se trouve actuellement à
                /*l'intérieur d'un mot. */


  do
    {
     printf("Nom du fichier texte : ");
     scanf("%s", NOM_FICH);
     PF = fopenp(NOM_FICH, "r");
      if (!PF)
         printf("Impossible d'ouvrir le fichier: %s.\n", NOM_FICH);
```

```
    }
  while (!PF);
  /* Initialisations des variables */
  for (I=0; I<26; I++)
  ABC[I]=0;
  NTOT    =0;
  NAUTRES =0;
  NMOTS   =0;
  DANS_MOT=0;
  /* Examination des caractères du fichier */
  while (!feof(PF))
     {
      C=fgetc(PF);
      if (!feof(PF))
         {
          /* Comptage au niveau caractères */
              NTOT++;
              if (C>='a' && C<='z')
                 ABC[C-'a']++;
              else if (C>='A' && C<='Z')
                 ABC[C-'A']++;
              else
                 NAUTRES++;



          /* Comptage des mots */
          if (F_SEPA(C))
             {
              if (DANS_MOT)
                 {
                  NMOTS++;
                  DANS_MOT=0;
                 }
             }
          else
             DANS_MOT=1;
         }
      }
   /* Fermeture du fichier */
  fclose(PF);
   /* Affichage du résultat */
  printf("Votre fichier contient :\n");
  printf("\t%d mots\n", NMOTS);
  printf("\t%d caractères\ndont\n", NTOT);
  for (I=0; I<26; I++)
       printf("\t%d fois la lettre %c\n", ABC[I], 'a'+I);
  printf("et %d autres caractères\n", NAUTRES);
  return 0;
}


int F_SEPA(char C)
{
 /* Tableau contenant tous les séparateurs de mots */
 char SEP[12] = { '\n', ' ', ',', ';', '.', ':','?', '!', '(', ')', '"', '\'' };
 int I;

 /* Comparaison de C avec tous les éléments du tableau */
 for (I=0 ; C!=SEP[I] && I<12 ; I++) ;
```

```
 if (I==12)
     return 0;
 else
     return 1;

}
```

Exercice 2

Soit le type structure suivant, représentant un point d'un plan :

struct point
{ char c ; // nom attribué au point
int x, y ; // ses coordonnées
}

Écrire une fonction qui reçoit en argument l'adresse d'une structure du type point et qui renvoie en résultat une structure de même type correspondant à un point de même nom et de coordonnées opposées.

Écrire un petit programme d'essai.

Solution

```
#include <stdio.h>

struct point
{ char c ;
int x, y ;
} ;

point sym (point * adp) /* la fonction qui renvoie la symétrie d'un point*/

 { point res ;
    res.c = adp->c ;
    res.x = - adp->x ;
    res.y = - adp->y ;
    return res ;
  }
```

```
Voici un exemple d'essai de notre fonction (ici, nous avons utilisé les
possibilités d'initialisation d'une structure pour donner des valeurs à p1) :

main()
{
point sym (point *) ;
point p1 = {'P', 5, 8} ;
point p2 ;
p2 = sym (&p1) ;
printf(" le point P1 %c%d%d\n", p1.c, p1.x, p1.y) ;
printf(" le point P2 %c%d%d\n", p2.c, p2.x, p2.y) ;

}
```

Université Hassan 1er
Faculté des Sciences et Techniques
Département Mathématiques et Informatique

Problème 1

Soit un fichier texte contenant des données organisées dans des enregistrements de longueur fixe.

	Code client	Nom	Age
Enrg1	12	LATIF	23
Enrg2	13	BELHAJ	56
Enrg3	23	FILALI	23

Ecrire un programme qui permet de gérer le contenu du fichier **client.txt** grâce au menu suivant :

1 : Ajouter un nouveau client
2 : Lister tous les clients
3 : Chercher un client
4 : Modifier le nom d'un client
5 : Supprimer un client
6 : Quitter

SOLUTION

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

void enregister(FILE* pf);
void afficher(FILE* pf);
void inserer(FILE* pf);
void modifier(FILE* p_ancien,char *n_nom);
void supprimer(FILE* p_ancien,char *n_nom);



int main()
{

FILE* fpointeur;

char n_nom[50];

int trouve;
int x=2;
int choix = 1;


while (choix)
{
printf("\n\n\n ======================== Menu ======================\n");
printf("|                                                        |\n");
printf("|                                                        |\n");
printf("| 1 :      ENREGISTRER DES NOUVEAUX ETUDIANTS            | \n");
printf("|                                                        |\n");
printf("| 2 :      AFFICHER        TOUS    LES ETUDIANTS         |\n");
```

```
printf("|                                                    |\n");
printf("| 3 :      INSERER UN ETUDIANT DANS FICHIER TRIE     |\n");
printf("|                                                    |\n");
printf("| 4 :      MODIFIER UN ETUDIANT                      |\n");
printf("|                                                    |\n");
printf("| 5 :      SUPPRIMER UN ETUDIANT                     |\n");
printf("|                                                    |\n");
printf("| O :     QUITTER                                    | \n");
printf("|                                                    |\n");
    printf(" ====================================================\n\n");
printf("                          VOTRE CHOIX : ");


scanf("%d",&choix);
printf("   %d",choix);

switch (choix)
{
case 1 :

   fpointeur =fopen("d:\\document.txt","a");
   if (fpointeur == NULL)
    {
     printf("Erruer d'ouverture\n");
     exit(-1);
    }
    enregister(fpointeur);  /* APPEL FUNCTIO  D AJOUT*/
    fclose(fpointeur);
    break;

case 2 :
    fpointeur =fopen("d:\\document.txt","r");
    afficher(fpointeur);  /* APPEL FUNCTIO  D AFFICHAGE*/
    break;

case 3 :
    fpointeur = fopen("d:\\document.txt","r");
    inserer(fpointeur);   /* APPEL FUNCTION  D INSERTION*/
 break;

case 4 :
   fpointeur = fopen("d:\\document.txt","r");
   printf(" donner le nom d'étudiant à modifier\n");
   scanf("%s",n_nom);
   modifier(fpointeur,n_nom); /* APPEL FUNCTION  DE MODIFICATION*/
 break;

 case 5 :
   fpointeur = fopen("d:\\document.txt","r");
   printf(" donner le nom d'étudiant à SUPPRIMER\n");
   scanf("%s",n_nom);
   supprimer(fpointeur,n_nom) ;/* APPEL FUNCTION  DE SUPPRESSION*/
 break;
default :
        exit(-1);
}
getch();
}

return (0);
```

```
}
/*Définition des 4 fonctions * /
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
struct personne
       {
          int mat ;
          char nom[50];
          int age;
        } prs,prs_n;
        FILE* p_ancien;
FILE* p_nouveau;
/****************** enregisTrer  UN nouveau ETUDIANT ***************/
void enregister(FILE* fpointeur)
{
char rep='O';
int i=1;
while (rep =='O')
  {
        printf("DONNER LES CORDONNEES DU NOUVEAU ETUDIANT \n");
        scanf("%d%s%d",&prs.mat,prs.nom,&prs.age);
        fprintf(fpointeur,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
        printf("Voulez vous  ajouter d'autre personne  O\\N ???\n");
        rep =getch();
        i++;
  }
}
void modifier(FILE* p_ancien,char *n_nom)
{
   p_nouveau = fopen("d:\\temp.txt","w");
   printf(" donner les nouvelles informations d'\202tudiant %s\n",n_nom);
   scanf( "%d %s  %d", &prs_n.mat,prs_n.nom,&prs_n.age);
 while(!feof(p_ancien))
   {
     fscanf(p_ancien,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     if (strcmp(prs.nom,n_nom) ==0)
     fprintf(p_nouveau,"%d %s  %d\n", prs_n.mat,prs_n.nom,prs_n.age);
     else
     fprintf(p_nouveau,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
   }
   fclose(p_nouveau);
   fclose(p_ancien);

   p_ancien = fopen("d:\\document.txt","w");
   p_nouveau = fopen("d:\\temp.txt","r");
   while(!feof(p_nouveau))
   {
     fscanf(p_nouveau,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     fprintf(p_ancien,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
}
  fclose(p_nouveau);
  fclose(p_ancien);

}
/***************************INSERER ***********************/
void inserer(FILE* p_ancien)
{
   int trouve= 0;
```

```
   printf(" donner les données du nouveau étudiant\n");
   scanf("%d%s%d",&prs_n.mat,prs_n.nom,&prs_n.age);
   printf("%-3d %-30s  %d\n", prs_n.mat,prs_n.nom,prs_n.age);

   p_nouveau = fopen("d:\\temp.txt","w");

   while(!feof(p_ancien)&& !trouve)
   {
     fscanf(p_ancien,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     if (strcmp(prs.nom,prs_n.nom) > 0) trouve  = 1;
     else
     fprintf(p_nouveau,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
   }
   fprintf(p_nouveau,"%d %s  %d\n", prs_n.mat,prs_n.nom,prs_n.age);
     if (trouve)  fprintf(p_nouveau,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
    while(!feof(p_ancien))
   {
       fscanf(p_ancien,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     fprintf(p_nouveau,"%d %s  %d\n", prs.mat,prs.nom,prs.age);

   }
   fclose(p_nouveau);
   fclose(p_ancien);

   p_ancien = fopen("d:\\document.txt","w");
   p_nouveau = fopen("d:\\temp.txt","r");
   while(!feof(p_nouveau))
   {
     fscanf(p_nouveau,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     fprintf(p_ancien,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
}
  fclose(p_nouveau);
  fclose(p_ancien);
}
/****************** SUPPRIMER UN ETUDIANT DONNE *****************/

void supprimer(FILE* p_ancien,char *n_nom)
{
  FILE* p_nouveau;
  struct personne prs;

   p_nouveau = fopen("d:\\temp.txt","w");

   /*strcpy(n_nom, "youssef");*/

   while(!feof(p_ancien))
   {
     fscanf(p_ancien,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     if (strcmp(prs.nom,n_nom)!=0)
          fprintf(p_nouveau,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
   }
   fclose(p_nouveau);
   fclose(p_ancien);

   p_ancien = fopen("d:\\document.txt","w");
   p_nouveau = fopen("d:\\temp.txt","r");
   while(!feof(p_nouveau))
   {
     fscanf(p_nouveau,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
     fprintf(p_ancien,"%d %s  %d\n", prs.mat,prs.nom,prs.age);
```

```
}
  fclose(p_nouveau);
  fclose(p_ancien);
}
/****************** FIN FUNCTION *****************/

/****************** AFFICHAGE DU CONTENU DU FICHIER *****************/

void afficher(FILE* fpointeur)
{
struct  personne prs;
if (fpointeur == NULL)
   {
    printf("Erruer d'ouverture\n");
    exit(-1);
   }

   printf("\n\nles cordonn\202es des \202tudiants enregistr\202s: \n\n");
  while (!feof(fpointeur))
  {

        fscanf(fpointeur,"%d%s%d\n",&prs.mat,prs.nom,&prs.age);
        printf("%-3d %-30s  %d\n", prs.mat,prs.nom,prs.age);

  }
    fclose(fpointeur);
}
/****************** FIN FUNCTION *****************/
```

Problème 2

Ecrire un programme qui vérifie la validité d'une série de numéros de comptes mémorisés dans un fichier(compte.txt). Un numéro de compte composé de trois parties: un numéro, un séparateur '-' et un numéro de contrôle. Un numéro de compte est correct:

- si le reste de la division entière de la valeur devant le séparateur '-' par 97 est différent de zéro et égal à la valeur de contrôle.

- si le reste de la division par 97 est zéro et la valeur de contrôle est 97.

Exemple:

Nombre de CCP 15742-28 : 15742 modulo 97 = 28 correct

Nombre de CCP 72270-5 : 72270 modulo 97 = 5 correct

Nombre de CCP 22610-10 : 22610 modulo 97 = 9 incorrect

Nombre de CCP 50537-0 : 50537 modulo 97 = 0
nombre incorrect, car la valeur de contrôle devrait être 97.
Nombre de CCP 50537-97 : 50537 modulo 97 = 0 correct

Utiliser une fonction Compte_TEST qui obtient comme paramètres les deux parties numériques d'un nombre de CCP et qui affiche alors un message indiquant si le numéro de compte est valide ou non.

Solution

```
#include <stdio.h>
#include <stdlib.h>
main()
{

  void Compte_TEST(long COMPTE, int CONTROLE);
   /* Déclarations : */
  /* Noms des fichiers et pointeurs de référence */
  char NOM_FICH[]  = "C:\\Compte.TXT";
  FILE *PF;

  long COMPTE;  /* nombre du compte  */
  int CONTROLE; /* nombre de contrôle   */

  /* Ouverture du fichier compte.TXT en lecture */
  PF = fopen(NOM_FICH, "r");
   if (!PF)
     {
      printf(" Impossible d'ouvrir le fichier: %s.\n", NOM_FICH);
      exit(-1);
     }

  while (!feof(PF)
     {
      fscanf (PF, "%ld-%d\n", &COMPTE, &CONTROLE);
      compte_TEST(COMPTE, CONTROLE);
     }
   /* Fermeture du fichier */
  fclose(PF);
   return 0;
}


void Compte_TEST(long COMPTE, int CONTROLE)
{
 int RESTE;
 RESTE = COMPTE % 97;
 if (RESTE == 0)
     RESTE = 97;
 if (RESTE == CONTROLE)
     printf ("Le nombre CCP %ld-%d est valide\n", COMPTE, CONTROLE);
 else
     printf ("Le nombre CCP %ld-%d n'est pas valide\n", COMPTE, CONTROLE);
}
```