Université Sidi Mohamed Ben Abdellah **Année Universitaire: 2014 / 2015**

Faculté des Sciences Dhar El Mahraz Fès **S3: SMI**

Département d'Informatique

Programmation I : Section Normale
Durée : 1 h 30 mn

Exercices 1

1) Ecrire un programme qui lit un fichier et compte le nombre d'occurrences de a, de b, de x, et de w. On utilisera l'instruction de branchement multiple ***switch.***

2) Soit P un pointeur qui 'pointe' sur un tableau A:

```
int A[] = {13, 24, 35, 46, 57, 68, 79, 90, 91};
int *P;
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions:

a) *P+3
b) *(P+3)
c) &P+2
d) &A[5]-4
e) A+4
f) &A[8]-P
g) P+(*P-10)
h) *(P+*(P+8)-A[7])

3) Ecrire une fonction ***Calcul***, en langage C, permettant d'effectuer les opérations de base d'une calculatrice. La fonction ***Calcul*** comprend trois variables passés en paramètre : ***deux entiers*** et un ***caractère*** qui représente l'opération à effectuer. La fonction retourne le résultat de l'opération demandé. les opérateurs à gérer sont: + (la somme), - (la soustraction), * (la multiplication) et / (la division).

NB : pensez à utiliser l'instruction switch … case

4) Ecrire le programme permettant de calculer le développement limité de sin(x) avec une précision ε :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - ... + \varepsilon(x)$$

Exercices 2

Le but de cet exercice est de réaliser un programme permettant de permuter les éléments d'un tableau :
Par exemple, si on considère le tableau suivant :

350	**56**	**78**	**20**	**2**	**13**	**180**

Voici l'ordre des éléments du tableau après permutation.

180	**13**	**2**	**20**	**78**	**56**	**350**

On a échanger **T[i]** avec **T[n-i-1]** où **n** est la taille du tableau.

1) Le programme doit réaliser les taches suivantes :
   a) Saisir la taille d'un tableau (maximum 30 éléments)
   b) Remplir le tableau par des entiers
   c) Permuter les éléments de ce tableau
   d) Afficher le tableau après permutation
2) Réécrire le programme précédent à l'aide des fonctions

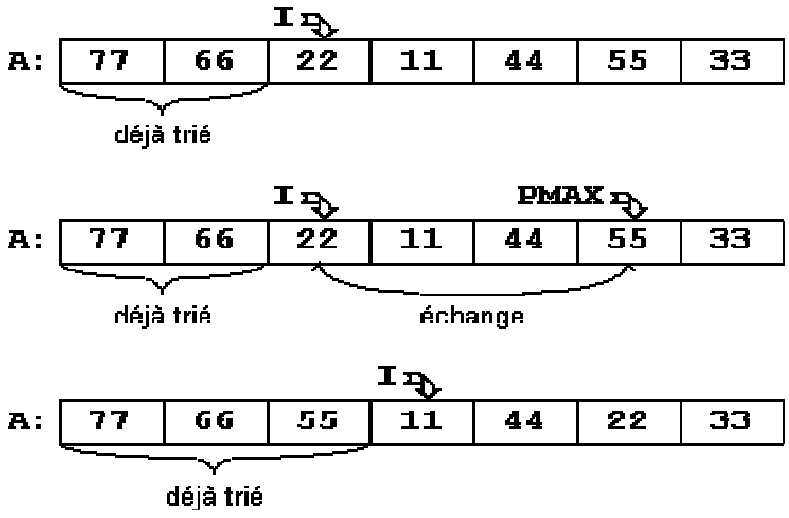
Exercices 3 : Tri par sélection du maximum

Problème: Classer les éléments d'un tableau A par ordre décroissant.

Méthode: Parcourir le tableau de gauche à droite à l'aide de l'indice I. Pour chaque élément A[I] du tableau, déterminer la position PMAX du (premier) maximum à droite de A[I] et échanger A[I] et A[PMAX].

NB : Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exemple:



Université Sidi Mohamed Ben Abdellah **Année Universitaire: 2014 / 2015**

Faculté des Sciences Dhar El Mahraz Fès **S3: SMI**

Département d'Informatique

Programmation I : Section Normale
Durée : 1 h 30 mn
CORRIGE

CORRIGE 1

1) Ecrire un programme qui lit un fichier et compte le nombre d'occurences de a, de e, de i, de o et de u. On utilisera l'instruction de branchement multiple switch.

```
/*** calcul du nombre d'occurences de a, e, i, o et u  ***/
#include <stdio.h>
main()
{
  char c;
  int nb_a = 0, nb_e = 0, nb_i = 0, nb_o = 0, nb_u = 0;

  while ((c = getchar()) != EOF)
    {
     switch (c)
       {
       case 'a':          nb_a++;          break;
       case 'e':          nb_e++;          break;
       case 'i':          nb_i++;          break;
       case 'o':          nb_o++;          break;
       case 'u':          nb_u++;          break;
       }
    }
  printf("Nombre de a = %d\n",nb_a);
  printf("Nombre de e = %d\n",nb_e);
  printf("Nombre de i = %d\n",nb_i);
  printf("Nombre de o = %d\n",nb_o);
  printf("Nombre de u = %d\n",nb_u);
}
```

2) Soit P un pointeur qui 'pointe' sur un tableau A:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int *P;
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions:

a) ***P+2** => la valeur 14

b) ***(P+2)** => la valeur 34

c) **&P+1** => l'adresse du pointeur derrière le pointeur P (rarement utilisée)

d) **&A[4]-3** => l'adresse de la composante A[1]

e) **A+3** => l'adresse de la composante A[3]

f) **&A[7]-P** => la valeur (indice) 7

g) **P+(*P-10)** => l'adresse de la composante A[2]

3) *int Calcul(int a, int b, char c) {*

```
switch(c){
case '+' : return(a + b);     break;
case '-' : return(a – b);     break;
case '*' : return(a * b);     break;
case '/' : return( a / b);    break;
        default :return( ???)
}
```

4)

```
#include <stdio.h>

main() {
double x, epsilon, result, precision ;
int n,fact;
x= ;
precision=  ;
result=x;
fact=3*2*1;
n=1;
epsilon=result*x*x/fact;
while (epsilon>precision)
        {
                result=result+epsilon;
                n+=2;
                fact=fact*n*(n-1);
                epsilon=epsilon*x*x/fact;
        }
}
```

CORRIGE 2

```
#include <stdlib.h>
#include <stdio.h>
int main ()
{
        int n, i, T[30], tmp;

        //saisie la taille
        do{
        scanf("%d", &n) ;
        while(n < 1 | | n > 30 ) ;

        //saisie des notes
        for (i = 0; i < n/2; i++)
                scanf("%d", &T[i]);
        //permutation
```

```
        for (i = 0; i < n; i++){
                tmp = T[i];
                T[i] = T[n – i – 1 ];
                T[n – i – 1 ] = tmp;
        }
        //affichage
        for (i = 0; i < n; i++)
                printf("%d", T[i]);
   return EXIT_SUCCESS;
}
```

CORRIGE 3

```
#include <stdio.h>
main() {
 /* Déclarations */
 int A[50]; /* tableau donné */
 int N;     /* dimension     */
 int I;     /* rang à partir duquel A n'est pas trié */
 int J;     /* indice courant      */
 int AIDE;  /* pour la permutation */
 int PMAX;  /* indique la position de l'élément */
          /* maximal à droite de A[I]         */
 /* Saisie des données */
 printf("Dimension du tableau (max.50) : ");
 scanf("%d", &N );
 for (J=0; J<N; J++)
    {
     printf("Elément %d : ", J);
     scanf("%d", &A[J]);
    }
 /* Affichage du tableau */
 printf("Tableau donné :\n");
 for (J=0; J<N; J++)
    printf("%d ", A[J]);
 printf("\n");
/* Tri du tableau par sélection directe du maximum. */
 for (I=0; I<N-1; I++)
     {
      /* Recherche du maximum à droite de A[I] */
      PMAX=I;
      for (J=I+1; J<N; J++)
         if (A[J]>A[PMAX]) PMAX=J;
      /* Echange de A[I] avec le maximum */
      AIDE=A[I];
      A[I]=A[PMAX];
      A[PMAX]=AIDE;
     }
  /* Edition du résultat */
 printf("Tableau trié :\n");
 for (J=0; J<N; J++)
     printf("%d ", A[J]);
 printf("\n");
 return 0;}
```