

Série N° 1
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

Ecrire un programme qui permet de saisir 3 notes d'un étudiant dans 3 matières, étant donnés les coefficients respectifs 3, 2 et 1, et de calculer sa moyenne générale.

Exercice 2 :

Ecrire un programme qui permet de permuter les valeurs de A et B sans utiliser de variable auxiliaire.

Exercice 3 :

Ecrire un programme qui permet de saisir un nombre réel puis de déterminer s'il appartient à un intervalle donné, sachant que les extrémités de l'intervalle sont fixées par l'utilisateur.

Exercice 4 :

Ecrire un programme qui permet de saisir un numéro de couleur de l'arc-en-ciel et d'afficher la couleur correspondante:

1: Rouge, 2: Orange, 3: Jaune, 4: Vert, 5: Bleu, 6: Indigo et 7: Violet

Exercice 5 :

Ecrire un programme qui permet de saisir les trois paramètres d'une équation de second degré, et de discuter les solutions selon les valeurs de a, b et c.

Exercice 6 :

Ecrire un programme qui lit un nombre N entier et qui affiche la somme et le produit de 1 à N:

1. En utilisant while,
2. En utilisant do... while,
3. En utilisant for.

Série N° 2
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

Ecrire un programme qui lit la dimension N d'un tableau T du type entier, remplit le tableau par des valeurs entrées au clavier. Copiez ensuite toutes les valeurs strictement positives dans un deuxième tableau et toutes les valeurs négatives dans un troisième tableau. Afficher les trois tableaux.

Exercice 2 :

Ecrire un programme qui transforme un nombre décimal saisi au clavier en binaire. Exemple $(10)_{10} = (1010)_2$.

Exercice 3 :

Ecrire un programme qui lit la dimension N d'un tableau T du type entier, remplit le tableau par des valeurs entrées au clavier.

1. Trier le tableau par la méthode de tri à bulle.
2. Rechercher dans le tableau T une valeur X entrée au clavier. Afficher la position de X si elle se trouve dans le tableau.

Exercice 4 :

Ecrire un programme qui met à zéro les éléments de la diagonale principale d'une matrice carrée A.

Exercice 5 :

Ecrire un programme qui réalise l'addition et la multiplication de deux matrices A et B.

Exercice 6 :

Ecrire un programme qui transforme une matrice de dimension (N,M) en un vecteur de dimension N*M.

Série N° 3
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

Ecrire un programme qui lit la dimension N d'un tableau T du type entier, remplit le tableau par des valeurs entières triées par ordre croissant entrées au clavier. Ensuite, insérer une valeur donnée au clavier dans le tableau T de manière à obtenir un tableau de valeurs triées. Utiliser le formalisme pointeur.

Exercice 2 :

Ecrire un programme qui lit la dimension N d'un tableau T du type entier, remplit le tableau par des valeurs entrées au clavier. Ensuite, effacer toutes les occurrences d'une valeur donnée au clavier dans le tableau T et tasser les éléments restants. Utiliser le formalisme pointeur.

Exercice 3 :

Ecrire un programme qui lit Les notes de N élèves d'une classe dans un devoir et les mémorise dans un tableau. Puis, en utilisant le formalisme pointeur, rechercher et afficher:

- La note maximale,
- La note minimale,
- La moyenne des notes.

Exercice 4 :

Ecrire un programme qui construit et affiche une matrice carrée unitaire de dimensions entrées au clavier. Utiliser le formalisme pointeur.

Exercice 5 :

Ecrire un programme qui réalise la multiplication d'une matrice de dimensions (N,M) et un vecteur de dimension M. utiliser le formalisme pointeur.

TD3 Informatique

Ex1

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

main()
{
    int *t, i, n, x;
    printf("Donner le nombre d'elt ");
    scanf("%d", &n);
    t = (int*) malloc( n * sizeof(int) );
    if (t == NULL) printf("Pas d'espace mémoire");
    for (i=0; i<n; i++) scanf("%d", &t[i]);
    printf("Saisie de la valeur de x");
    scanf("%d", &x);
    for (i=n; i>0 && (t[i-1] > x; i--))
        *(t+i) = *(t+i-1);
    *(t+i) = x;
    for (i=0; i<=n; i++) printf("%d\t", *(t+i));
    free(t);
    getch();
}
```


Ex2

Lire la dimension du tableau n

Allocation ^{dynamique} pour n élt

* Saisir le tableau et ^{on lie} savoir quelle valeur qu'il répète et la placer d'ici

```
for (i=0; i<n; i++)  
    if (*t[i] != x)  
        {  
            *(t+j) = *t[i];  
            j++;  
        }
```

```
for (i=0; i<j; i++)  
    printf("%d\t", *t[i]);
```

Ex4

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

int *t, n;

printf("Entrer l'ordre de la matrice : ");

scanf("%d", &n);

t = (int *) malloc (n * sizeof(int));

if (t == NULL) printf("Espace indisponible");

for (int i=0; i<n; i++)

t[i] = (int *) malloc (n * sizeof(int));

if (t[i] == NULL) printf("Espace indisponible");

for (i=0; i<n; i++)

for (j=0; j<n; j++)

if (i == j) *t[i][j] = 1;

else *t[i][j] = 0;


```

for (i=0; i<n; i++)
{
    f(int j=0; j<n; j++)
        printf("%d", *((int*)t + i*n+j));
    printf("\n");
}

```

```

for (i=0; i<n; i++)
    free(t[i]);
free(t);
getch();
}

```


Série N° 4
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

Ecrire un programme qui lit une chaîne de caractères CH et qui convertit toutes les majuscules dans des minuscules et vice-versa. Le résultat sera mémorisé dans la même variable CH et affiché après la conversion.

1. Utiliser le formalisme tableau de caractères.
2. Utiliser le formalisme pointeur.

Exercice 2 :

Ecrire un programme qui lit un texte TXT et qui enlève toutes les apparitions d'un caractère entré au clavier en tassant les éléments restants. Les modifications se feront dans la même variable TXT.

1. Utiliser le formalisme tableau de caractères.
2. Utiliser le formalisme pointeur.

Exercice 3 :

Ecrire un programme qui lit un verbe régulier en "er" au clavier et qui en affiche la conjugaison au présent de l'indicatif de ce verbe. Contrôlez s'il s'agit bien d'un verbe en "er" avant de conjuguer.

1. Utiliser le formalisme tableau de caractères.
2. Utiliser le formalisme pointeur.
3. Utiliser les fonction strcpy, strlen.

Exercice 4 :

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. Utilisez la méthode de tri par sélection.

1. Utiliser le formalisme pointeur de pointeurs.

Série N° 5
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

Ecrire un programme qui permet d'effectuer les opérations suivantes : (On n'utilisera que les fonctions)

- Saisir les éléments d'un tableau T d'entiers
- Trier le tableau
- Afficher le tableau trié.
- Afficher la position dans le tableau, d'une valeur saisie au clavier si elle appartient au tableau T, sinon afficher que l'élément n'existe pas.

N.B. :

Le programme doit définir au moins les fonctions citées ci-dessous. Vous les définissez dans un fichier appelés "malib.h" . Dans votre programme principal "prog.c", inclure le fichier "malib.h"

Exercice 2 :

Ecrire un programme qui permet d'effectuer le produit de deux matrices carées :(On n'utilisera que les fonctions)

- Saisir les éléments de la matrice A et B d'entiers
- Afficher les trois matrices A et B et C.

Exercice 3 :

Ecrire un programme qui lit 10 mots et les mémorise dans un tableau de chaînes de caractères : on n'utilisera que les fonctions)

- Saisir les mots dans un tableau de chaînes de caractères
- Afficher les mots avant le tri
- Trier les mots lexicographiquement .
- Afficher le tableau trié.

Série N° 6
Module Informatique III
Cycle Préparatoire (S3)

Exercice 1 :

La liste des étudiants est représentée par un tableau de structures qu'on va allouer dynamiquement.

Un étudiant est représenté par un enregistrement composé des champs suivant :

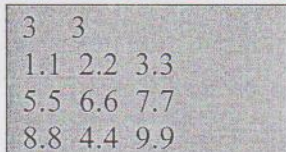
- **Nom** : chaîne de caractères représentant le nom de l'étudiant,
- **NI** : entier représentant le numéro de l'étudiant,
- **Groupe** : caractère représentant le groupe auquel appartient cet étudiant.

Donnez un programme pour (On n'utilisera que les fonctions) :

- Saisir la liste des étudiants,
- Afficher cette liste,
- Trier, dans l'ordre croissant des noms, le tableau de structures
- Afficher le tableau trié,
- Un étudiant veut rechercher son numéro d'inscription, pour cela, il tape au clavier son nom et votre programme doit afficher son nom, son numéro d'inscription et le groupe auquel il appartient.

Exercice 2 :

1. Créer à l'aide d'un éditeur de texte le fichier texte "**matrice.txt**" sous la forme suivante:



```
3 3
1.1 2.2 3.3
5.5 6.6 7.7
8.8 4.4 9.9
```

2. Lire ce fichier texte et faire une allocation dynamique pour une matrice.
3. Remplir les éléments de cette matrice avec les informations du fichier texte.
4. Afficher les éléments de la matrice.
5. Ecrire un programme qui met à zéro les éléments de la diagonale principale d'une matrice.
6. Ecrire les éléments de la matrice après diagonalisation dans un nouveau fichier "matriceR.txt".



Université Ibn Tofail
Ecole Nationale des Sciences
Appliquées, Kénitra.

Année Universitaire 2012/2013

Contrôle Continu
Module Informatique III
Cycle Préparatoire (S3)
2 Heures

Exercice 1 :

Ecrire un programme en C qui calcule et affiche la valeur de π au moyen de la série suivante (100 termes) :

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

Exercice 2 :

Ecrire un programme qui calcule la racine carrée d'un nombre réel positif A en utilisant la suite de Newton.

$$X_0 = A$$

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{A}{X_n} \right)$$

Le calcul s'arrête quand $|X_{n+1} - X_n| < \xi$ avec ξ donné

Nous posons : $Y = X_{n+1}$ et $X = X_n$

Exercice 3 :

Soit la suite suivante :

$$U_n = n^2 - 5n \quad \text{avec} \quad n \in [1, N]$$

Ecrire un programme en utilisant le formalisme pointeur permettant de :

- Remplir un tableau U par les termes de la suite U_n .
- Afficher les termes de la suite.
- Supprimer les valeurs négatives de U_n dans le tableau.
- Afficher les termes restants de la suite.

Exercice 4 :

Ecrire un programme en utilisant le formalisme pointeur de pointeur qui réalise la multiplication de deux matrices A et B. Le résultat de la multiplication sera mémorisé dans une troisième matrice C qui sera ensuite affichée.

(float) i/j \rightarrow ix(1.0)/j

Corrigé du contrôle

Ex 1

1^{ère} méthode

```
#include <conio.h>
```

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

```
main()
```

```
{ float pi, s;
```

```
  int N;
```

```
  printf("Donner le nombre de terme : ");
```

```
  scanf("%d", &N);
```

```
   $s=0$ ;  
  for(int i=1; i<N; i++)
```

```
  { if(i%2 == 0) s += (float) 1/(2*i+1);
```

```
    else s += -1/(2*i+1);
```

```
  }
```

```
  pi = 4*(1+s);
```

```
  printf("La valeur de pi est %f", pi);
```

```
  getch();
```

2^{ème} méthode

```
  k = 1;
```

```
  s = 0;
```

```
  for(i=0; i<N; i++)
```

```
  { s += k*x(1.0)/(2*i+1);
```

```
    k = -k;
```

```
  }
```

Ex 2

Y = A

do

```
{ X = Y;
```

```
  Y = (X + A/X)/2;
```



```

val = y - x;
if (val < 0) val = -val;
while (val >= eps);

```

```

#include <conio.h>
#include <stdio.h>
#define eps 0.01.

```

```

main()

```

```

{
    float x, y, val;

```

```

    int A;

```

```

    scanf("%d", &A);

```

```

    y = A;

```

```

    do
    {

```

```

        x = y;

```

```

        y = 0.5 * (x + A/x);

```

```

        val = y - x;

```

```

        if (y - x < 0)

```

```

            val = -val;

```

```

    } while (val >= eps);

```

```

    printf("La racine carrée de %d est %f", A, y);

```

```

    getch();

```

```

}

```

Ex3

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

#include <conio.h>

```

```

main()

```

```

{

```



```

int * t, n, j = 0;
printf("Veuillez saisir le nombre de termes de la suite");
scanf("%d", &n);
t = (int*) malloc (n * sizeof(int));
if (t == 0) printf("Pas d'espace mémoire");
for (int i = 0; i < n; i++)
    *(t+i) = (i+1)*(i+1) - 5*(i+1);
for (int i = 0; i < n; i++)
    printf("%d\t", *(t+i));
for (int i = 0; i < n; i++)
    if (*(t+i) >= 0)
        {
            *(t+j) = *(t+i);
            j++;
        }
for (int i = 0; i < j; i++)
    printf("%d\t", *(t+i));
free(t);
getch();
}

```

Ex 4

```

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

main()
{
    int **A, **B, **C, M, L, N, K;
    printf("Donner le nbr de ligne de la matrice A");
    scanf("%d", &L);
    printf("Donner le nbr de colonne de la matrice A");
    scanf("%d", &M);
}

```


printf("Donner le nombre de ligne de la matrice B");

scanf("%d", &N);

printf("Donner le nombre de colonne de la matrice B");

scanf("%d", &K);

A = (int**) malloc(L * sizeof(int*));

if (A == 0) printf("pas d'espace mémoire");

for (int i = 0; i < L; i++) { A[i] = (int*) malloc(M * sizeof(int));

if (A[i] == 0) printf("pas d'espace mémoire");

de m pr B $L \rightarrow N$ $M \rightarrow K$

de m pr C $L \rightarrow L$ $M \rightarrow K$.

printf("Veuillez saisir les élt de la matrice A");

for (int i = 0; i < L; i++)

for (int j = 0; j < M; j++)

scanf("%d", (int*) A + i * M + j)

de m pr B $L \rightarrow N$ $M \rightarrow K$.

mettre zéro C

if (M == ~~N~~)

{ for (int i = 0; i < L; i++)

for (int j = 0; j < M; j++)

for (int k = 0; k < K; k++)

* ((int*) C + i * K + k) += * ((int*) A + i * M + j) * ((int*) B + j * K + k);

for (int i = 0; i < L; i++)

for (int k = 0; k < K; k++)

printf("%d\t", * ((int*) C + i * K + k));

Libérer l'espace mémoire

2.



Contrôle Terminal
Module Informatique III
Cycle Préparatoire (S3)
2 Heurs

Exercice 1 :

On souhaite créer un programme d'annuaire téléphonique très simplifié qui associe à un nom de personne un numéro de téléphone.

Créer une structure *Personne* pouvant contenir ces informations (nom, prénom, date de naissance et *Num_téléphone*).

La date de naissance est elle-même une structure comportant trois champs correspondant au jour, au mois et l'année.

L'annuaire est représenté par un tableau de structures qu'on va allouer dynamiquement.

Ecrire un programme en utilisant que des fonctions pour:

1. Saisir les informations de chaque personne,
2. Afficher ces informations,
3. Trier, dans l'ordre croissant des noms, ce tableau de structures
4. Afficher le tableau trié,
5. Un utilisateur veut chercher le numéro de téléphone d'une personne, pour cela, il tape au clavier le nom de la personne et votre programme doit afficher le nom, le numéro de téléphone et la date de naissance.

Exercice 2 :

1. Supposons que le fichier texte "**vecteur.txt**" sous la forme suivante:

1	3
1.1	2.2 3.3
1	3
4.4	5.5 6.6

2. Lire ce fichier texte et faire une allocation dynamique pour chaque vecteur.
3. Remplir les éléments de ces vecteurs avec les informations du fichier texte
4. Afficher les éléments de chaque vecteur
5. Ecrire un programme qui réalise le produit scalaire des deux vecteurs.

LES ENTRÉES-SORTIES CONVERSATIONELLES

format	conversion en	écriture
%d	int	décimale signée
%ld	long int	décimale signée
%u	unsigned int	décimale non signée
%lu	unsigned long int	décimale non signée
%o	unsigned int octale	non signée
%lo	unsigned long int	octale non signée
%x	unsigned int hexadécimale	non signée
%lx	unsigned long int hexadécimale	non signée
%f	double décimale	virgule fixe
%lf	long double	décimale virgule fixe
%e	double	décimale notation exponentielle
%le	long double décimale	notation exponentielle
%g	double décimale, représentation la plus courte parmi %f et %e	
%lg	long double décimale, représentation la plus courte parmi %lf et %le	
%c	unsigned char	caractère
%s	char*	chaîne de caractères

LES TYPES CARACTÈRES:

NOTATION EN C	CODE ASCII (hexadécimal)	ABREVIATION USUELLE
\a	07	cloche ou bip (alert ou audible bell)
\b	08	Retour arrière (Backspace)
\f	0C	Saut de page (Form Feed)
\n	0A	Saut de Ligne (Line Feed)
\r	0D	Retour chariot (Carriage Return)
\t	09	Tabulation horizontale (Horizontal Tab)
\v	0B	Tabulation verticale (Vertical Tab)