

COMPTE RENDU MISE À NIVEAU ALGORITHMIQUE  
GIMENEZ FLORIAN - M1 EEEA

# Contents

<b>1 Travaux Pratique 1</b>	<b>3</b>
1.1 Programme affichant "Bonjour" . . . . .	3
1.1.1 Code . . . . .	3
1.1.2 Résultat . . . . .	3
1.2 Calcul Aire d'un disque . . . . .	3
1.2.1 Code . . . . .	3
1.2.2 Résultat . . . . .	4
1.3 Signe du produit . . . . .	4
1.3.1 Code . . . . .	4
1.3.2 Résultat . . . . .	4
1.4 Polynôme . . . . .	5
1.4.1 Code . . . . .	5
1.4.2 Résultat . . . . .	5
1.5 Conversion de secondes vers le format HH:MM:SS . . . . .	6
1.5.1 Code . . . . .	6
1.5.2 Résultat . . . . .	6
1.6 Calculatrice . . . . .	7
1.6.1 Code . . . . .	7
1.6.2 Résultat . . . . .	8
<b>2 Travaux Pratique 2</b>	<b>8</b>
2.1 Factorielle . . . . .	8
2.1.1 Code . . . . .	8
2.1.2 Résultat . . . . .	9
2.2 Calcul de l'exponentiel . . . . .	9
2.2.1 Approximation de e . . . . .	9
2.2.2 Approximation de e, avec un condition sur la précision . .	9
2.3 Calcul de $f(x)$ . . . . .	10
2.3.1 Code . . . . .	10
2.3.2 Résultat . . . . .	10
2.4 Inversion chiffres d'un nombre . . . . .	11
2.4.1 Code . . . . .	11
2.4.2 Résultat . . . . .	11
2.5 Calcul d'un nombre de visiteurs . . . . .	12
2.5.1 Code . . . . .	12
2.5.2 Résultat . . . . .	12
<b>3 Travaux Pratique 3</b>	<b>21</b>
3.1 Etude d'un tableau . . . . .	21
3.1.1 Code . . . . .	21
3.1.2 Résultat . . . . .	21
3.2 Etude d'une Matrice . . . . .	21
3.2.1 Code . . . . .	21
3.2.2 Résultat . . . . .	22

<b>4</b>	<b>Travaux Pratique 4</b>	<b>22</b>
4.1	Exercice 1 . . . . .	22
4.1.1	Question 2 . . . . .	22
4.1.2	Question 3 : Affichage ordre croissant et décroissant . . .	23
4.1.3	Code . . . . .	23
4.1.4	Résultat . . . . .	23
4.2	Exercice 2 . . . . .	24
4.2.1	Code . . . . .	24
4.2.2	Résultat . . . . .	24
4.3	Exercice 3 . . . . .	25
4.3.1	Code . . . . .	25
4.3.2	Résultat . . . . .	25
4.4	Exercice 4 . . . . .	26
4.4.1	Code . . . . .	26
4.4.2	Résultat . . . . .	26

# 1 Travaux Pratique 1

## 1.1 Programme affichant "Bonjour"

### 1.1.1 Code

```
#include <stdlib.h>
#include <stdio.h>
void main()
{
    printf("Hello World");
}
```

Figure 1: Programme C pour le programme Hello World

Pour afficher "Hello World" nous devons simplement placer un *printf*.

### 1.1.2 Résultat

```
./GIMENEZ_bonjour
Hello World
```

Figure 2: Résultat du Programme *bonjour*

## 1.2 Calcul Aire d'un disque

### 1.2.1 Code

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

int main()
{
    int D;
    float pi = 3.14;

    printf ("Quel est le diamètre du disque ?\n");
    //D prend la valeur qu'entre l'utilisateur
    scanf("%d",&D);
    //On a : Aire = 2*pi*R². On effectue directement le calcul dans le printf pour faire l'économie d'une variable
    printf("La surface d'un cercle de rayon %d est : %.3f\n",D,(pi*D*D)/4);
    return 0;
}
```

Figure 3: Programme C pour le calcul de l'aire d'un Disque

Ici nous avons demandé le diamètre d'un disque pour pouvoir calculer son aire. Nous avons aussi affiché le résultat avec 3 chiffres après la virgule. Nous avons aussi fait le calcul directement dans le *printf*, pour faire l'économie d'une variable et rendre le programme plus lisible.

### 1.2.2 Résultat

```
./GIMENEZ disque
Quel est le diamètre du disque ?
2
La surface d'un cercle de rayon 2 est : 3.140
```

Figure 4: Résultat du Programme *disque*

## 1.3 Signe du produit

### 1.3.1 Code

```
#include <stdio.h>
int main()
{
    float a,b;
    //On entre les deux nombres dont on veut connaître le signe du produit
    printf ("Quel est le premier nombre ?\n");
    scanf ("%f",&a);
    printf ("Quel est le deuxième nombre ?\n");
    scanf ("%f",&b);

    //Traitement du cas particulier du produit nul
    if((a==0) || (b==0))
        printf("Produit nul\n");
    else
    {
        //Si A est positif le signe dépendra de b
        if (a>0)
        {
            if(b>0)
                printf("Produit positif\n");
            else
                printf("Produit négatif\n");
        }
        //Si A est négatif le signe dépend toujours de b mais les conditions sont inversée
        else
        {
            if(b>0)
                printf("Produit négatif\n");
            else
                printf("Produit positif\n");
        }
    }
    return 0;
}
```

Figure 5: Programme C pour la détermination du signe d'un produit

Nous cherchons à déterminer si le signe résultant du produit de deux nombres sera positif ou négatif. Pour cela nous testons tout d'abord le cas particulier où l'un des deux nombres est nul. Par la suite en fonction du signe des nombres on pourra déterminer le signe du résultat.

### 1.3.2 Résultat

```
./GIMENEZ produit
Quel est le premier nombre ?
1
Quel est le deuxième nombre ?
-1
Produit négatif
```

Figure 6: Résultat du Programme *produit*

## 1.4 Polynôme

### 1.4.1 Code

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

int main()
{
    float x;
    //Saisi d'une valeur à donner à la variable pour déterminer f(x)
    printf ("Entrer un réel pour calculer la valeur de f(x) ?\n");
    scanf("%f",&x);
    //On calcul f(x) directement dans le printf, et on affiche une précision de 3 chiffres
    //après la virgule
    printf("f(%.3f) est : %.3f\n",x,3*pow(x,4)+2*pow(x,3)+pow(x,2)-x-2);

    return 0;
}
```

Figure 7: Programme C pour le calcul d'une fonction

Nous souhaitons créer un programme permettant de calculer la valeur de la fonction  $f$  pour un  $x$  donné. Nous avons utilisé la fonction *power* de la librairie `math.h`, pour effectuer les calculs de puissance.

### 1.4.2 Résultat

```
./GIMENEZ polynomes
Entrer un réel pour calculer la valeur de f(x) ?
2
f(2.000) est : 64.000
```

Figure 8: Résultat du Programme *polynome*

## 1.5 Conversion de secondes vers le format HH:MM:SS

### 1.5.1 Code

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

int main()
{
    int sec_init;
    int sec,min,hour;

    //On entre le nombre de secondes que l'on souhaite convertir
    printf("Entrer le nombre de secondes à convertir \n");
    scanf("%d",&sec_init);

    //On calcule combien d'heure cela représente
    hour = sec_init/3600;
    //On soustrait le nombre d'heure pour trouver le nombre de minutes qu'il reste à convertir
    min = (sec_init-(3600*hour))/60;
    //Enfin on soustrait le nombre de minutes et d'heure pour obtenir les secondes restantes
    sec = sec_init-((60*min)+(3600*hour));
    //On affiche le résultat sous le format HH:MM:SS
    printf("%d:%d:%d",hour,min,sec);

    return 0;
}
```

Figure 9: Programme C pour la conversion d'un nombre de secondes en nombres d'heures, minutes et secondes

Nous avons ici créer un programme qui prend en entrées un nombre de secondes et qui l'a converti en nombres d'heures, de minutes, et de secondes. On notera que pour ce programme toute les divisions sont des divisions entière où on laisse le soin au compilateur d'arrondir les valeurs.

### 1.5.2 Résultat

```
//GIMENEZ conversion
Entrer le nombre de secondes à convertir
13604
3:46:44
```

Figure 10: Résultat du Programme *conversion*

## 1.6 Calculatrice

### 1.6.1 Code

```
int main()
{
    float A,B,R;
    int op, choix = 1;
    printf("Bienvenue sur le programme de calcul\n");
    //On boucle tant que l'utilisateur souhaite effectuer des calculs
    while (choix == 1)
    {
        //Entrée du premier nombre
        printf("Entrer un nombre\n");
        scanf("%f",&A);
        //Choix de l'opération
        printf("Quel opération souhaitez vous faire? (+1 ou *-2)\n");
        scanf("%d",&op);
        //Si l'utilisateur à commis une erreur sur son choix on n'effectue pas le calcul
        if((op == 1) || (op == 2))
        {
            //Entrée du deuxième nombre
            printf("Entrer un deuxième nombre\n");
            scanf("%f",&B);
            //On calcul A * B ou A + B en fonction du choix de l'utilisateur et on affiche le résultat
            if (op == 1) {R = A + B; printf ("%f + %f = %f\n",A,B,R); }
            else {R = A * B; printf ("%f * %f = %f\n",A,B,R); }
        }
        //Si l'utilisateur s'est trompé on lui indique
        else printf("Erreur sur le choix...\n");
        //On demande si l'utilisateur souhaite continuer à utiliser le programme ou non
        printf("Souhaitez vous faire une autre opération? (1 = oui)\n");
        scanf("%d",&choix);
    }
    return 0;
}
```

Figure 11: Programme C pour la création d'une calculatrice

Ici nous avons créé un programme qui nous permet d'entrer deux nombres et d'effectuer une opération mathématique (+ ou \*). Nous demandons alors à l'utilisateur d'entrer un premier nombre puis un opération et enfin un deuxième nombre. On vérifie que l'utilisateur à bien un opération valable.



### 1.6.2 Résultat

```
//GIMENEZ_calculatrice
Bienvenue sur le programme de calcul
Entrer un nombre
5
Quel opération souhaitez vous faire? (+-1 ou *-2)
1
Entrer un deuxième nombre
12
5.000 + 12.000 = 17.000
Souhaitez vous faire une autre opération? (1 = oui)
1
Entrer un nombre
14
Quel opération souhaitez vous faire? (+-1 ou *-2)
0
Erreur sur le choix..
Souhaitez vous faire une autre opération? (1 = oui)
0
```

Figure 12: Résultat du Programme *calculatrice*

## 2 Travaux Pratique 2

### 2.1 Factorielle

#### 2.1.1 Code

```
int factorielle (int n)
{
    int i, R;
    //On initialise la variable résultat à 1 qui l'élément neutre du produit
    R = 1;
    for (i=2; i <= n; i++)
        R = R * i;
    return R;
}

int main()
{
    int i, max;
    max = 10;

    for (i = 0; i <= max; i++)
    {
        printf("Le resultat de %d! = %d\n",i,factorielle(i));
    }

    return 0;
}
```

Figure 13: Programme C du calcul de la facotrielle

Le but de ce code était de pouvoir calculer la factorielle d'un nombre et aussi d'afficher les résultats intermédiaires. On a ici créer une fonction pour faciliter la construction du texte. On a donc plus qu'à récupérer les valeurs et les afficher. La construction de la fonction fait que pour 0!, on obtient bien 1 comme résultat.

### 2.1.2 Résultat

```
./GIMENEZ_factorielle
Le resultat de 0! = 1
Le resultat de 1! = 1
Le resultat de 2! = 2
Le resultat de 3! = 6
Le resultat de 4! = 24
Le resultat de 5! = 120
Le resultat de 6! = 720
Le resultat de 7! = 5040
Le resultat de 8! = 40320
Le resultat de 9! = 362880
Le resultat de 10! = 3628800
```

Figure 14: Résultat du Programme *factorielle*

## 2.2 Calcul de l'exponentiel

### 2.2.1 Approximation de e

```
int factorielle (int n)
{
    int i, R;
    R = 1;
    for (i=2; i <= n; i++)
        R = R * i;
    return R;
}

double exp_approx (float x, int n)
{
    double R;
    int i;
    R = 1.0;
    for (i=1; i <= n; i++)
    {
        R = R + 1.0/factorielle(i);
    }
    return R;
}

int main()
{
    float x=1;
    int n;

    printf("Entrer une valeur de n pour la précision du calcul de e\n");
    scanf("%d",&n);

    printf("exp(%.3f) pour n = %d nous donne : %f\n",x,n,exp_approx(x,n));
    return 0;
}
```

Figure 15: Programme C du calcul de e

On a ici créé un programme qui nous permet d'approximer la valeur de  $e$  pour une précision  $n$  donnée.

```
./GIMENEZ_exponentiel
Entrer une valeur de n pour la précision du calcul de e
5
exp(1.000) pour n = 5 nous donne : 2.716667
```

Figure 16: Résultat du Programme pour  $n = 5$

### 2.2.2 Approximation de e, avec un condition sur la précision

On souhaite maintenant créer un programme qui nous donne un  $n$  qui obéit aux la règles suivantes :  $\frac{1}{(1-n)!} > 10^{-4}$  et  $\frac{1}{n!} < 10^{-4}$

```

int estim_n()
{
    int i=0;
    double est;
    do
    {
        //On incrémente i jusqu'à que la première conditions soit vérifié
        i++;
        est = 1./factorielle(i);
    }while ( est > 0.0001 );

    //On teste si la deuxième condition est vérifié
    if ( (1./factorielle(i-1)) > 0.0001)
        return i;

    //Si la condition n'est pas vérifié on renvoie une erreur
    else
    {
        printf("!Erreur Condition!");
        return -1;
    }
}

```

Figure 17: Programme C qui donne  $n$  vérifiant notre condition

```

./GIMENEZ exponentiel2
n qui vérifie la condition est : 8
exp(1.000) pour n = 8 nous donne : 2.718254

```

Figure 18: Résultat du Programme qui donne  $n = 8$

## 2.3 Calcul de $f(x)$

### 2.3.1 Code

```

double f_x (float x)
{
    if (x > 2)
        return 100;
    else if ( x > 1)
        return exp(x);
    else if ( x > 0)
        return sin(x);
    else return pow(x,2);
}

int main()
{
    float x;
    int choix = 1;
    do
    {
        printf("Entrer un réel pour obtenir f(x)\n");
        scanf("%f",&x);
        printf("Pour x = %.3f on a f(%.3f) = %.3f\n",x,x,f_x(x));
        printf("Entrer 1 pour continuer\n");
        scanf("%d",&choix);
    } while(choix == 1);
    return 0;
}

```

Figure 19: Programme C définition de notre fonction  $f(x)$

On souhaite maintenant créer un programme qui nous donne calcul l'image de la fonction  $f$  donnée. Pour cela nous avons défini chaque condition et renvoyons en fonction du  $x$  donnée le résultat correspondant.

### 2.3.2 Résultat

```

./GIMENEZ f x
Entrer un réel pour obtenir f(x)
-0.5
Pour x = -0.500 on a f(-0.500) = 0.250
Entrer 1 pour continuer
1
Entrer un réel pour obtenir f(x)
1
Pour x = 1.000 on a f(1.000) = 0.841
Entrer 1 pour continuer
1
Entrer un réel pour obtenir f(x)
1.5
Pour x = 1.500 on a f(1.500) = 4.482
Entrer 1 pour continuer
1
Entrer un réel pour obtenir f(x)
100
Pour x = 100.000 on a f(100.000) = 100.000

```

Figure 20: Résultat du Programme pour les x données dans l'énoncé

## 2.4 Inversion chiffres d'un nombre

### 2.4.1 Code

```

int swap_ (int x)
{
    int a,b,c,d;

    //Condition pour que notre programme soit fonctionnel
    if (x > 9999)
    {
        printf("Nombre trop grand!");
        return -1;
    }
    //Xmodulo10 nous donne le dernier nombre de X puis on divise par 10 et on réitère ainsi on récupère la centaine
    //puis les dizaines et etc.
    a = x%10;
    x = x/10;

    b = x%10;
    x = x/10;

    c = x%10;
    x = x/10;

    d = x%10;

    //On teste la longueur de notre nombre pour afficher notre nombres inversé
    if (d != 0)
    {
        return (a*1000+b*100+c*10+d);
    }
    else if ((c != 0) && (d == 0))
    {
        return (a*100+b*10+c);
    }
    else return (a*10 + b);
}

```

Figure 21: Programme C

On crée maintenant un programme permettant d'inverser l'entier entré au clavier par l'utilisateur, par exemple si on prend le nombre 845, le programme doit nous renvoyer la valeur 548.

### 2.4.2 Résultat

```

./GIMENEZ swap_figures
Entrer un entier à inverser
-8645
x inverser est : -5468

```

Figure 22: Résultat du Programme avec un nombre négatif

## 2.5 Calcul d'un nombre de visiteurs

### 2.5.1 Code

```
int main()
{
    int nb_entrees = 30;
    int recette_obj = 50;
    int t_retraite = 2;
    int t_enfant = 1;
    int t_adulte = 3;

    int nb_adultes=0, nb_enfants=0, nb_retraites=0;
    int recette=0, nb_visiteur=0;

    printf("Nous remplissons les conditions pour les combinaisons suivante :\n\n");
    //On teste chaque combinaisons possible
    for (nb_enfants = 0; nb_enfants < 30 ; nb_enfants++)
        for (nb_retraites = 0; nb_retraites < 30; nb_retraites++)
            for (nb_adultes = 0; nb_adultes < 30 ; nb_adultes++)
            {
                nb_visiteur = nb_adultes + nb_retraites + nb_enfants;
                recette = t_adulte * nb_adultes + t_retraite * nb_retraites + t_enfant * nb_enfants;
                if((nb_visiteur == nb_entrees) && (recette == recette_obj))
                {
                    printf("-----\n");
                    printf("Nombre d'adultes : %d, Nombre de retraités : %d, Nombre d'enfants : %d\n",nb_adultes
                        ,nb_retraites,nb_enfants);
                }
            }
    return 0;
}
```

Figure 23: Programme C

On souhaite connaître toutes les possibilités de visiteurs qui respectent les conditions suivantes :

Le total des visiteurs d'un parc était de 30 personnes. Les tarifs appliqués sont définis selon les catégories : 3 euros pour un adulte, 2 euros pour un retraité et 1 euros pour un enfant. La recette des entrées était de 50 euros.

### 2.5.2 Résultat

```
./nb_visiteurs
Nous remplissons les conditions pour les combinaisons suivante :
-----
Nombre d'adultes : 16, Nombre de retraités : 1, Nombre d'enfants : 0
-----
Nombre d'adultes : 14, Nombre de retraités : 4, Nombre d'enfants : 0
-----
Nombre d'adultes : 12, Nombre de retraités : 7, Nombre d'enfants : 0
-----
Nombre d'adultes : 10, Nombre de retraités : 10, Nombre d'enfants : 0
-----
Nombre d'adultes : 8, Nombre de retraités : 13, Nombre d'enfants : 0
-----
Nombre d'adultes : 6, Nombre de retraités : 16, Nombre d'enfants : 0
-----
Nombre d'adultes : 4, Nombre de retraités : 19, Nombre d'enfants : 0
-----
Nombre d'adultes : 2, Nombre de retraités : 22, Nombre d'enfants : 0
-----
Nombre d'adultes : 0, Nombre de retraités : 25, Nombre d'enfants : 0
-----
Nombre d'adultes : 15, Nombre de retraités : 2, Nombre d'enfants : 1
```

Figure 24: Exemple de résultat

Ici nous avons trop de lignes résultats pour mettre en photo les voici donc ci dessous :

Nous remplissons les conditions pour les combinaisons suivantes :

---

Nombre d'adultes : 16, Nombre de retraités : 1, Nombre d'enfants : 0

---

Nombre d'adultes : 14, Nombre de retraités : 4, Nombre d'enfants : 0

---

Nombre d'adultes : 12, Nombre de retraités : 7, Nombre d'enfants : 0

---

Nombre d'adultes : 10, Nombre de retraités : 10, Nombre d'enfants : 0

---

Nombre d'adultes : 8, Nombre de retraités : 13, Nombre d'enfants : 0

---

Nombre d'adultes : 6, Nombre de retraités : 16, Nombre d'enfants : 0

---

Nombre d'adultes : 4, Nombre de retraités : 19, Nombre d'enfants : 0

---

Nombre d'adultes : 2, Nombre de retraités : 22, Nombre d'enfants : 0

---

Nombre d'adultes : 0, Nombre de retraités : 25, Nombre d'enfants : 0

---

Nombre d'adultes : 15, Nombre de retraités : 2, Nombre d'enfants : 1

---

Nombre d'adultes : 13, Nombre de retraités : 5, Nombre d'enfants : 1

---

Nombre d'adultes : 11, Nombre de retraités : 8, Nombre d'enfants : 1

---

Nombre d'adultes : 9, Nombre de retraités : 11, Nombre d'enfants : 1

---

Nombre d'adultes : 7, Nombre de retraités : 14, Nombre d'enfants : 1

---

Nombre d'adultes : 5, Nombre de retraités : 17, Nombre d'enfants : 1

---

Nombre d'adultes : 3, Nombre de retraités : 20, Nombre d'enfants : 1

---

Nombre d'adultes : 1, Nombre de retraités : 23, Nombre d'enfants : 1

---

Nombre d'adultes : 16, Nombre de retraités : 0, Nombre d'enfants : 2

---

Nombre d'adultes : 14, Nombre de retraités : 3, Nombre d'enfants : 2

---

Nombre d'adultes : 12, Nombre de retraités : 6, Nombre d'enfants : 2

---

Nombre d'adultes : 10, Nombre de retraités : 9, Nombre d'enfants : 2

---

Nombre d'adultes : 8, Nombre de retraités : 12, Nombre d'enfants : 2

---

Nombre d'adultes : 6, Nombre de retraités : 15, Nombre d'enfants : 2

---

Nombre d'adultes : 4, Nombre de retraités : 18, Nombre d'enfants : 2

---

Nombre d'adultes : 2, Nombre de retraités : 21, Nombre d'enfants : 2

---

Nombre d'adultes : 0, Nombre de retraités : 24, Nombre d'enfants : 2

---

Nombre d'adultes : 15, Nombre de retraités : 1, Nombre d'enfants : 3

---

Nombre d'adultes : 13, Nombre de retraités : 4, Nombre d'enfants : 3

---

Nombre d'adultes : 11, Nombre de retraités : 7, Nombre d'enfants : 3

---

Nombre d'adultes : 9, Nombre de retraités : 10, Nombre d'enfants : 3

---

Nombre d'adultes : 7, Nombre de retraités : 13, Nombre d'enfants : 3

---

Nombre d'adultes : 5, Nombre de retraités : 16, Nombre d'enfants : 3

---

Nombre d'adultes : 3, Nombre de retraités : 19, Nombre d'enfants : 3

---

Nombre d'adultes : 1, Nombre de retraités : 22, Nombre d'enfants : 3

---

Nombre d'adultes : 14, Nombre de retraités : 2, Nombre d'enfants : 4

---

Nombre d'adultes : 12, Nombre de retraités : 5, Nombre d'enfants : 4

---

Nombre d'adultes : 10, Nombre de retraités : 8, Nombre d'enfants : 4

---

Nombre d'adultes : 8, Nombre de retraités : 11, Nombre d'enfants : 4

---

Nombre d'adultes : 6, Nombre de retraités : 14, Nombre d'enfants : 4

---

Nombre d'adultes : 4, Nombre de retraités : 17, Nombre d'enfants : 4

---

Nombre d'adultes : 2, Nombre de retraités : 20, Nombre d'enfants : 4

---

Nombre d'adultes : 0, Nombre de retraités : 23, Nombre d'enfants : 4

---

Nombre d'adultes : 15, Nombre de retraités : 0, Nombre d'enfants : 5

---

Nombre d'adultes : 13, Nombre de retraités : 3, Nombre d'enfants : 5

---

Nombre d'adultes : 11, Nombre de retraités : 6, Nombre d'enfants : 5

---

Nombre d'adultes : 9, Nombre de retraités : 9, Nombre d'enfants : 5

---

Nombre d'adultes : 7, Nombre de retraités : 12, Nombre d'enfants : 5

---

Nombre d'adultes : 5, Nombre de retraités : 15, Nombre d'enfants : 5

---

Nombre d'adultes : 3, Nombre de retraités : 18, Nombre d'enfants : 5

---

Nombre d'adultes : 1, Nombre de retraités : 21, Nombre d'enfants : 5

---

Nombre d'adultes : 14, Nombre de retraités : 1, Nombre d'enfants : 6

---

Nombre d'adultes : 12, Nombre de retraités : 4, Nombre d'enfants : 6

---

Nombre d'adultes : 10, Nombre de retraités : 7, Nombre d'enfants : 6

---

Nombre d'adultes : 8, Nombre de retraités : 10, Nombre d'enfants : 6

---

Nombre d'adultes : 6, Nombre de retraités : 13, Nombre d'enfants : 6

---

Nombre d'adultes : 4, Nombre de retraités : 16, Nombre d'enfants : 6

---

Nombre d'adultes : 2, Nombre de retraités : 19, Nombre d'enfants : 6

---

Nombre d'adultes : 0, Nombre de retraités : 22, Nombre d'enfants : 6

---

Nombre d'adultes : 13, Nombre de retraités : 2, Nombre d'enfants : 7

---

Nombre d'adultes : 11, Nombre de retraités : 5, Nombre d'enfants : 7

---

Nombre d'adultes : 9, Nombre de retraités : 8, Nombre d'enfants : 7

---

Nombre d'adultes : 7, Nombre de retraités : 11, Nombre d'enfants : 7

---

Nombre d'adultes : 5, Nombre de retraités : 14, Nombre d'enfants : 7

---

Nombre d'adultes : 3, Nombre de retraités : 17, Nombre d'enfants : 7

---

Nombre d'adultes : 1, Nombre de retraités : 20, Nombre d'enfants : 7

---

Nombre d'adultes : 14, Nombre de retraités : 0, Nombre d'enfants : 8

---

Nombre d'adultes : 12, Nombre de retraités : 3, Nombre d'enfants : 8

---

Nombre d'adultes : 10, Nombre de retraités : 6, Nombre d'enfants : 8

---

Nombre d'adultes : 8, Nombre de retraités : 9, Nombre d'enfants : 8

---

Nombre d'adultes : 6, Nombre de retraités : 12, Nombre d'enfants : 8

---

Nombre d'adultes : 4, Nombre de retraités : 15, Nombre d'enfants : 8

---

Nombre d'adultes : 2, Nombre de retraités : 18, Nombre d'enfants : 8

---

Nombre d'adultes : 0, Nombre de retraités : 21, Nombre d'enfants : 8

---

Nombre d'adultes : 13, Nombre de retraités : 1, Nombre d'enfants : 9

---

Nombre d'adultes : 11, Nombre de retraités : 4, Nombre d'enfants : 9



---

Nombre d'adultes : 9, Nombre de retraités : 7, Nombre d'enfants : 9

---

Nombre d'adultes : 7, Nombre de retraités : 10, Nombre d'enfants : 9

---

Nombre d'adultes : 5, Nombre de retraités : 13, Nombre d'enfants : 9

---

Nombre d'adultes : 3, Nombre de retraités : 16, Nombre d'enfants : 9

---

Nombre d'adultes : 1, Nombre de retraités : 19, Nombre d'enfants : 9

---

Nombre d'adultes : 12, Nombre de retraités : 2, Nombre d'enfants : 10

---

Nombre d'adultes : 10, Nombre de retraités : 5, Nombre d'enfants : 10

---

Nombre d'adultes : 8, Nombre de retraités : 8, Nombre d'enfants : 10

---

Nombre d'adultes : 6, Nombre de retraités : 11, Nombre d'enfants : 10

---

Nombre d'adultes : 4, Nombre de retraités : 14, Nombre d'enfants : 10

---

Nombre d'adultes : 2, Nombre de retraités : 17, Nombre d'enfants : 10

---

Nombre d'adultes : 0, Nombre de retraités : 20, Nombre d'enfants : 10

---

Nombre d'adultes : 13, Nombre de retraités : 0, Nombre d'enfants : 11

---

Nombre d'adultes : 11, Nombre de retraités : 3, Nombre d'enfants : 11

---

Nombre d'adultes : 9, Nombre de retraités : 6, Nombre d'enfants : 11

---

Nombre d'adultes : 7, Nombre de retraités : 9, Nombre d'enfants : 11

---

Nombre d'adultes : 5, Nombre de retraités : 12, Nombre d'enfants : 11

---

Nombre d'adultes : 3, Nombre de retraités : 15, Nombre d'enfants : 11

---

Nombre d'adultes : 1, Nombre de retraités : 18, Nombre d'enfants : 11

---

Nombre d'adultes : 12, Nombre de retraités : 1, Nombre d'enfants : 12

---

Nombre d'adultes : 10, Nombre de retraités : 4, Nombre d'enfants : 12

---

Nombre d'adultes : 8, Nombre de retraités : 7, Nombre d'enfants : 12

---

Nombre d'adultes : 6, Nombre de retraités : 10, Nombre d'enfants : 12

---

Nombre d'adultes : 4, Nombre de retraités : 13, Nombre d'enfants : 12

---

Nombre d'adultes : 2, Nombre de retraités : 16, Nombre d'enfants : 12

---

Nombre d'adultes : 0, Nombre de retraités : 19, Nombre d'enfants : 12

---

Nombre d'adultes : 11, Nombre de retraités : 2, Nombre d'enfants : 13

---

Nombre d'adultes : 9, Nombre de retraités : 5, Nombre d'enfants : 13

---

Nombre d'adultes : 7, Nombre de retraités : 8, Nombre d'enfants : 13

---

Nombre d'adultes : 5, Nombre de retraités : 11, Nombre d'enfants : 13

---

Nombre d'adultes : 3, Nombre de retraités : 14, Nombre d'enfants : 13

---

Nombre d'adultes : 1, Nombre de retraités : 17, Nombre d'enfants : 13

---

Nombre d'adultes : 12, Nombre de retraités : 0, Nombre d'enfants : 14

---

Nombre d'adultes : 10, Nombre de retraités : 3, Nombre d'enfants : 14

---

Nombre d'adultes : 8, Nombre de retraités : 6, Nombre d'enfants : 14

---

Nombre d'adultes : 6, Nombre de retraités : 9, Nombre d'enfants : 14

---

Nombre d'adultes : 4, Nombre de retraités : 12, Nombre d'enfants : 14

---

Nombre d'adultes : 2, Nombre de retraités : 15, Nombre d'enfants : 14

---

Nombre d'adultes : 0, Nombre de retraités : 18, Nombre d'enfants : 14

---

Nombre d'adultes : 11, Nombre de retraités : 1, Nombre d'enfants : 15

---

Nombre d'adultes : 9, Nombre de retraités : 4, Nombre d'enfants : 15

---

Nombre d'adultes : 7, Nombre de retraités : 7, Nombre d'enfants : 15

---

Nombre d'adultes : 5, Nombre de retraités : 10, Nombre d'enfants : 15

---

Nombre d'adultes : 3, Nombre de retraités : 13, Nombre d'enfants : 15

---

Nombre d'adultes : 1, Nombre de retraités : 16, Nombre d'enfants : 15

---

Nombre d'adultes : 10, Nombre de retraités : 2, Nombre d'enfants : 16

---

Nombre d'adultes : 8, Nombre de retraités : 5, Nombre d'enfants : 16

---

Nombre d'adultes : 6, Nombre de retraités : 8, Nombre d'enfants : 16

---

Nombre d'adultes : 4, Nombre de retraités : 11, Nombre d'enfants : 16

---

Nombre d'adultes : 2, Nombre de retraités : 14, Nombre d'enfants : 16

---

Nombre d'adultes : 0, Nombre de retraités : 17, Nombre d'enfants : 16

---

Nombre d'adultes : 11, Nombre de retraités : 0, Nombre d'enfants : 17

---

Nombre d'adultes : 9, Nombre de retraités : 3, Nombre d'enfants : 17

---

Nombre d'adultes : 7, Nombre de retraités : 6, Nombre d'enfants : 17

---

Nombre d'adultes : 5, Nombre de retraités : 9, Nombre d'enfants : 17

---

Nombre d'adultes : 3, Nombre de retraités : 12, Nombre d'enfants : 17

---

Nombre d'adultes : 1, Nombre de retraités : 15, Nombre d'enfants : 17

---

Nombre d'adultes : 10, Nombre de retraités : 1, Nombre d'enfants : 18

---

Nombre d'adultes : 8, Nombre de retraités : 4, Nombre d'enfants : 18

---

Nombre d'adultes : 6, Nombre de retraités : 7, Nombre d'enfants : 18

---

Nombre d'adultes : 4, Nombre de retraités : 10, Nombre d'enfants : 18

---

Nombre d'adultes : 2, Nombre de retraités : 13, Nombre d'enfants : 18

---

Nombre d'adultes : 0, Nombre de retraités : 16, Nombre d'enfants : 18

---

Nombre d'adultes : 9, Nombre de retraités : 2, Nombre d'enfants : 19

---

Nombre d'adultes : 7, Nombre de retraités : 5, Nombre d'enfants : 19

---

Nombre d'adultes : 5, Nombre de retraités : 8, Nombre d'enfants : 19

---

Nombre d'adultes : 3, Nombre de retraités : 11, Nombre d'enfants : 19

---

Nombre d'adultes : 1, Nombre de retraités : 14, Nombre d'enfants : 19

---

Nombre d'adultes : 10, Nombre de retraités : 0, Nombre d'enfants : 20

---

Nombre d'adultes : 8, Nombre de retraités : 3, Nombre d'enfants : 20

---

Nombre d'adultes : 6, Nombre de retraités : 6, Nombre d'enfants : 20

---

Nombre d'adultes : 4, Nombre de retraités : 9, Nombre d'enfants : 20

---

Nombre d'adultes : 2, Nombre de retraités : 12, Nombre d'enfants : 20

---

Nombre d'adultes : 0, Nombre de retraités : 15, Nombre d'enfants : 20

---

Nombre d'adultes : 9, Nombre de retraités : 1, Nombre d'enfants : 21

---

Nombre d'adultes : 7, Nombre de retraités : 4, Nombre d'enfants : 21

---

Nombre d'adultes : 5, Nombre de retraités : 7, Nombre d'enfants : 21

---

Nombre d'adultes : 3, Nombre de retraités : 10, Nombre d'enfants : 21

---

Nombre d'adultes : 1, Nombre de retraités : 13, Nombre d'enfants : 21

---

Nombre d'adultes : 8, Nombre de retraités : 2, Nombre d'enfants : 22

---

Nombre d'adultes : 6, Nombre de retraités : 5, Nombre d'enfants : 22

---

Nombre d'adultes : 4, Nombre de retraités : 8, Nombre d'enfants : 22

---

Nombre d'adultes : 2, Nombre de retraités : 11, Nombre d'enfants : 22

---

Nombre d'adultes : 0, Nombre de retraités : 14, Nombre d'enfants : 22

---

Nombre d'adultes : 9, Nombre de retraités : 0, Nombre d'enfants : 23

---

Nombre d'adultes : 7, Nombre de retraités : 3, Nombre d'enfants : 23

---

Nombre d'adultes : 5, Nombre de retraités : 6, Nombre d'enfants : 23

---

Nombre d'adultes : 3, Nombre de retraités : 9, Nombre d'enfants : 23

---

Nombre d'adultes : 1, Nombre de retraités : 12, Nombre d'enfants : 23

---

Nombre d'adultes : 8, Nombre de retraités : 1, Nombre d'enfants : 24

---

Nombre d'adultes : 6, Nombre de retraités : 4, Nombre d'enfants : 24

---

Nombre d'adultes : 4, Nombre de retraités : 7, Nombre d'enfants : 24

---

Nombre d'adultes : 2, Nombre de retraités : 10, Nombre d'enfants : 24

---

Nombre d'adultes : 0, Nombre de retraités : 13, Nombre d'enfants : 24

---

Nombre d'adultes : 7, Nombre de retraités : 2, Nombre d'enfants : 25

---

Nombre d'adultes : 5, Nombre de retraités : 5, Nombre d'enfants : 25

---

Nombre d'adultes : 3, Nombre de retraités : 8, Nombre d'enfants : 25

---

Nombre d'adultes : 1, Nombre de retraités : 11, Nombre d'enfants : 25

---

Nombre d'adultes : 8, Nombre de retraités : 0, Nombre d'enfants : 26

---

Nombre d'adultes : 6, Nombre de retraités : 3, Nombre d'enfants : 26

---

Nombre d'adultes : 4, Nombre de retraités : 6, Nombre d'enfants : 26

---

Nombre d'adultes : 2, Nombre de retraités : 9, Nombre d'enfants : 26

---

Nombre d'adultes : 0, Nombre de retraités : 12, Nombre d'enfants : 26

---

Nombre d'adultes : 7, Nombre de retraités : 1, Nombre d'enfants : 27

---

Nombre d'adultes : 5, Nombre de retraités : 4, Nombre d'enfants : 27

---

Nombre d'adultes : 3, Nombre de retraités : 7, Nombre d'enfants : 27

---

Nombre d'adultes : 1, Nombre de retraités : 10, Nombre d'enfants : 27

---

Nombre d'adultes : 6, Nombre de retraités : 2, Nombre d'enfants : 28

---

Nombre d'adultes : 4, Nombre de retraités : 5, Nombre d'enfants : 28

---

Nombre d'adultes : 2, Nombre de retraités : 8, Nombre d'enfants : 28

---

Nombre d'adultes : 0, Nombre de retraités : 11, Nombre d'enfants : 28

---

Nombre d'adultes : 7, Nombre de retraités : 0, Nombre d'enfants : 29

---

Nombre d'adultes : 5, Nombre de retraités : 3, Nombre d'enfants : 29

---

Nombre d'adultes : 3, Nombre de retraités : 6, Nombre d'enfants : 29

---

Nombre d'adultes : 1, Nombre de retraités : 9, Nombre d'enfants : 29

## 3 Travaux Pratique 3

### 3.1 Etude d'un tableau

#### 3.1.1 Code

```
int main (void)
{
    int T[10] = { 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    int i, S=0,min,max;

    //T[i] = i
    for ( i = 0; i < 10; ++i )
        T[i] = i*i;

    //Affichage du tableau
    for ( i = 0; i < 10; ++i )
    {
        printf ("%i", T[i]);
        printf ("\n");
    }

    //Moyenne
    for ( i = 0; i < 10; ++i )
        S += T[i];
    printf ("La moyenne est : %f\n", S/10.);

    //On initialise les deux variables avec les deux valeurs qui se trouvent au début du tableau
    min = T[0];
    max = T[0];
    //Min et Max
    for ( i = 0; i < 10; ++i )
    {
        if (T[i] < min) min = T[i];
        if (T[i] > max) max = T[i];
    }
    printf("Le Max de ce tableau est : %d\n Le Min de ce tableau est : %d\n",max,min);
    return 0;
}
```

Figure 25: Programme C

Depuis le code nous avons ajouté le remplissage du tableau de tel sorte à avoir  $T[i] = i^2$  sur l'ensemble de ce dernier. On a aussi implémenter la recherche du maximum, du minimum et de la moyenne.

#### 3.1.2 Résultat

```
./min_max
0
1
4
9
16
25
36
49
64
81
La moyenne est : 28.5
Le Max de ce tableau est : 81
Le Min de ce tableau est : 0
```

Figure 26: Résultat

## 3.2 Etude d'une Matrice

#### 3.2.1 Code

Ici nous avons entré la matrice en dur dans le code puis chercher le minimum et le maximum et enfin par un jeu d'indice de tableau nous affichons les cases que l'utilisateur souhaite voir.

```

int main (void)
{
    int T[4][5] = {{10,5,-3,17,82},{9,0,0,8,-7},{32,20,1,0,14},{0,0,8,7,6}};
    int c,l,min,max,choix=1;

    for ( l = 0; l < 4; ++l )
    {
        for ( c = 0; c < 5; ++c )
        {
            printf("%d ",T[l][c]);
        }
        printf("\n");
    }

    //Initialisation de Min et Max
    min = T[0][0];
    max = T[0][0];

    for ( l = 0; l < 4; ++l )
    {
        for ( c = 0; c < 5; ++c )
        {
            if (T[l][c] < min) min = T[l][c];
            if (T[l][c] > max) max = T[l][c];
        }
    }

    printf("Le Max de la matrice est : %d\n Le Min de la matrice est : %d\n",max,min);

    while (choix)
    {
        printf("Choix de la ligne : \n");
        scanf("%d",&l);
        printf("Choix de la colonne : \n");
        scanf("%d",&c);
        printf("La valeur de pour la ligne %d et la colonne %d est : %d\n",l,c,T[l][c]);
        printf("Voulez vous trouver une autre valeur? oui-non (1-0)\n");
        scanf("%d",&choix);
    }

    return 0;
}

```

Figure 27: Programme C

### 3.2.2 Résultat

```

./GIMENEZ matrice
10 5 -3 17 82
9 0 0 8 -7
32 20 1 0 14
0 0 8 7 6
Le Max de la matrice est : 82
Le Min de la matrice est : -7
Choix de la ligne :
2
Choix de la colonne :
4
La valeur de pour la ligne 2 et la colonne 4 est : 14
Voulez vous trouver une autre valeur? oui-non (1-0)
0

```

Figure 28: Résultat

## 4 Travaux Pratique 4

### 4.1 Exercice 1

#### 4.1.1 Question 2

Ici la variable p est un pointeur sur le tableau. Au premier printf, p nous donne la valeur de la première case. Lorsqu'on incrémente P on va pouvoir afficher les cases suivantes du tableaux.

### 4.1.2 Question 3 : Affichage ordre croissant et décroissant

#### 4.1.3 Code

```
int main()
{
    int tab[5] = {1, 2, 6, 0, 7};
    int *p;
    int i;
    p = tab;
    printf("**p = %d\n", *p);
    p += 1;
    printf("**p = %d\n", *p);

    p = tab;

    //Affichage par ordre croissant
    printf("Affichage par ordre croissant d'indices : \n");
    for (i = 0 ; i < 5 ; i++)
    {
        printf("**p = %d\n", *p);
        p += 1;
    }

    //Affichage par ordre décroissant
    printf("Affichage par ordre décroissant d'indices : \n");
    for (i = 4 ; i >= 0 ; i--)
    {
        p -= 1;
        printf("**p = %d\n", *p);
    }

    return 0;
}
```

Figure 29: Programme C

On applique ce que nous a appris la question précédente pour obtenir l’affichage du tableau dans l’ordre croissant des indices puis décroissant.

#### 4.1.4 Résultat

```
./ex1
*p = 1
*p = 2
Affichage par ordre croissant d'indices :
*p = 1
*p = 2
*p = 6
*p = 0
*p = 7
Affichage par ordre décroissant d'indices :
*p = 7
*p = 0
*p = 6
*p = 2
*p = 1
```

Figure 30: Résultat



## 4.2 Exercice 2

### 4.2.1 Code

```
void echange(int* p1, int* p2)
{
    int temp;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

int main()
{
    int a=1, b=2;
    printf("a = %d, b = %d\n",a,b);
    printf("Appelle de la fonction echange..\n");
    echange(&a,&b);
    printf("a = %d, b = %d\n",a,b);
    return 0;
}
```

Figure 31: Programme C

On utilise ici les propriétés propres aux pointeurs pour permettre un échange de valeurs entre deux variables.

### 4.2.2 Résultat

```
./ex2
a = 1, b = 2
Appelle de la fonction echange..
a = 2, b = 1
```

Figure 32: Résultat

## 4.3 Exercice 3

### 4.3.1 Code

```
int main()
{
    int tab[5] = {1, 2, 6, 0, 7};
    int tab2[5] = {1, 2, 6, 0, 7};
    int *p,*q;
    int *a,*b;
    int i;
    p = tab;
    q = tab;

    printf("Affichage du tableau initial : \n");
    for (i = 0 ; i < 5 ; i++)
    {
        printf("**p = %d\n",*q);
        q += 1;
    }
    q -= 1;

    printf("Affichage du tableau \"retourné\" : \n");
    for (i = 0 ; i < 5 ; i++)
    {
        a=*p;
        b=*q;
        exchange(&a,&b);
        tab2[i] = a;
        q -= 1;
        p += 1;

        printf("**p = %d\n",tab2[i]);
    }
    return 0;
}
```

Figure 33: Programme C

Ici nous utilisons la fonction défini précédemment pour inverser le tableau.

### 4.3.2 Résultat

```
./ex3
Affichage du tableau initial :
*p = 1
*p = 2
*p = 6
*p = 0
*p = 7
Affichage du tableau retourné :
*p = 7
*p = 0
*p = 6
*p = 2
*p = 1
```

Figure 34: Résultat

## 4.4 Exercice 4

### 4.4.1 Code

```
int main()
{
    int a=3, b=1, c=0;

    if((a > b))
        exchange(&a, &b);
    if((a > c))
        exchange(&a, &c);
    if (b > c)
        exchange(&b, &c);

    printf(" %d < %d < %d\n", a, b, c);

    return 0;
}
```

Figure 35: Programme C

Ici nous utilisons la fonction défini précédemment pour échanger les valeurs jusqu'à que les valeurs soient trié par ordre croissant.

### 4.4.2 Résultat

```
./ex4
0 < 1 < 3
```

Figure 36: Résultat