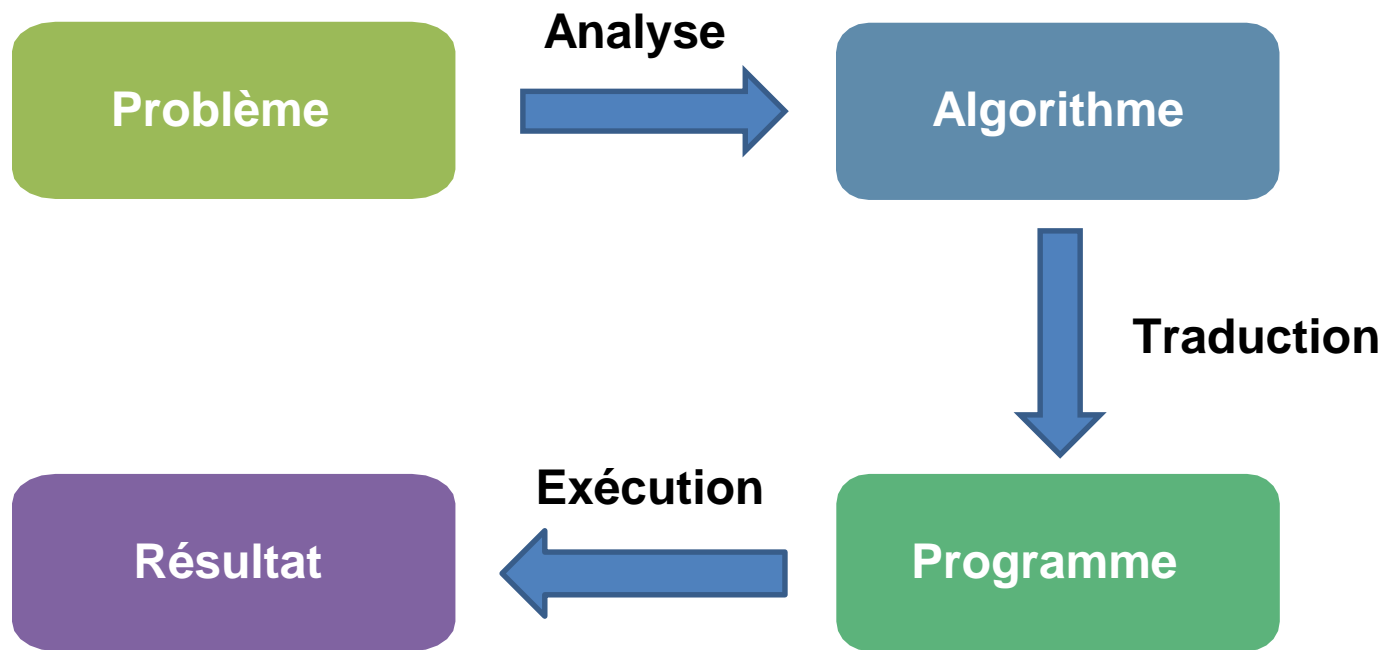


# Programmation en langage C

# Notion de programme

- Schéma de résolution d'un problème informatique



# Notion de programme

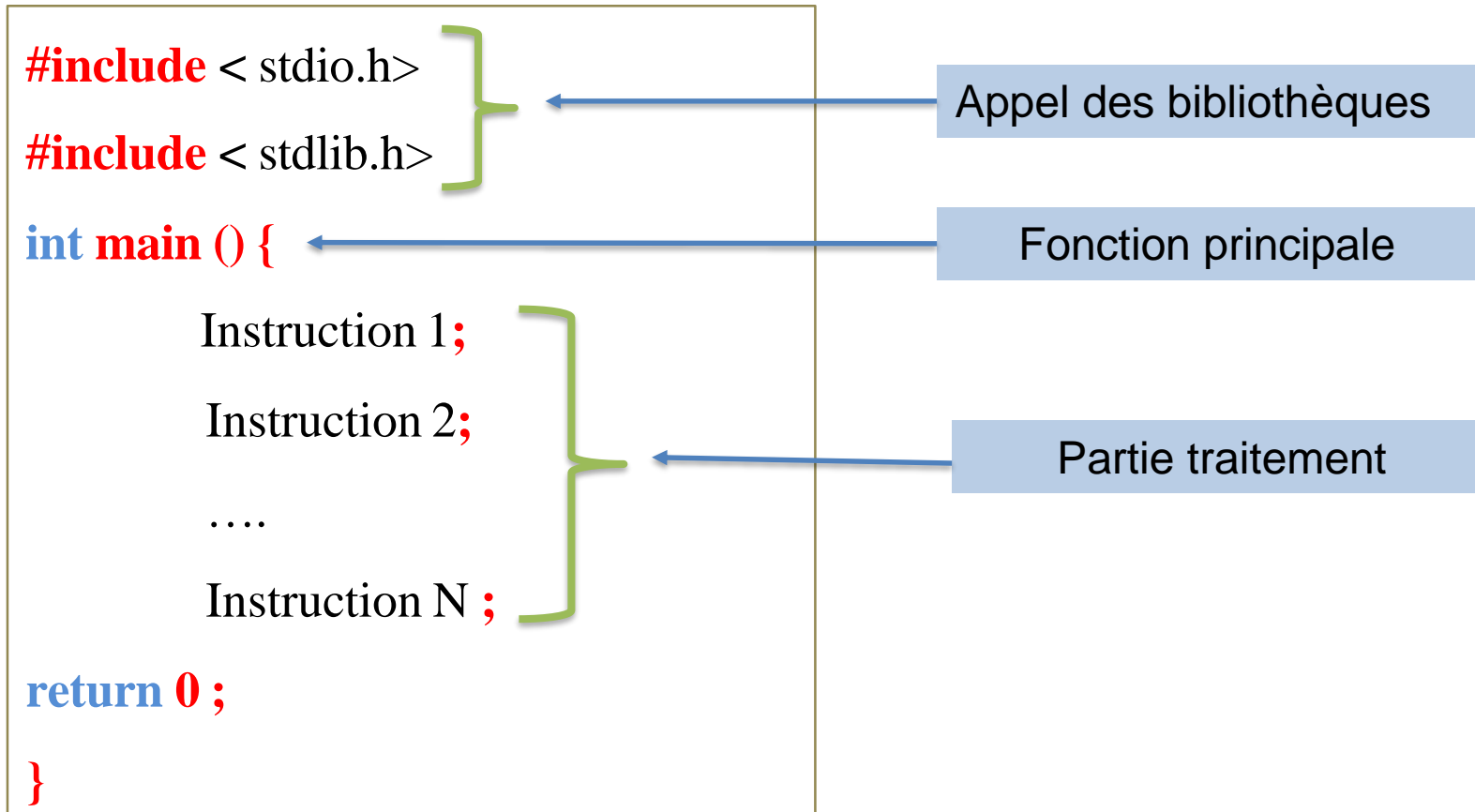
---

- **Notion de programme**

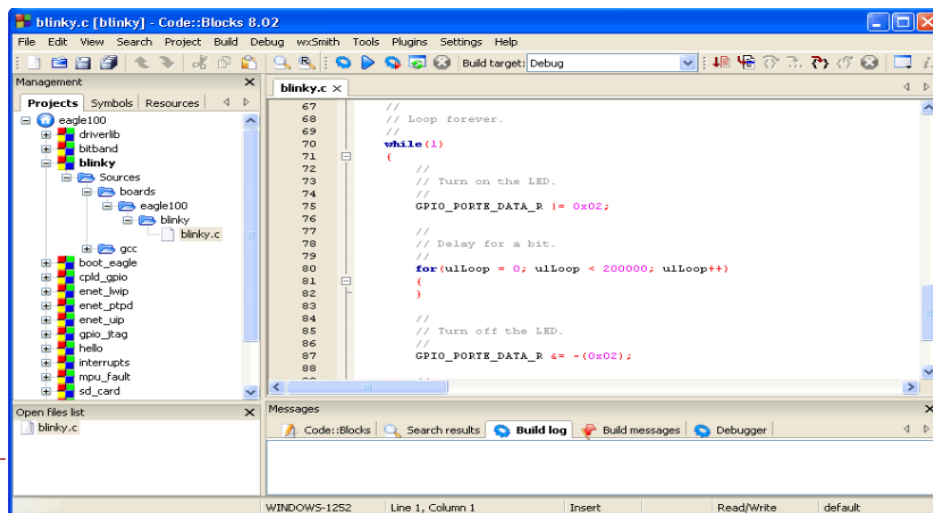
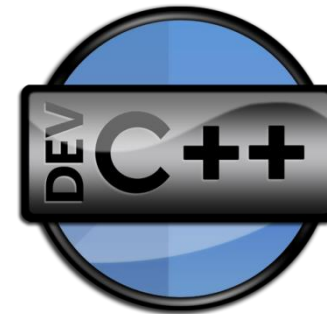
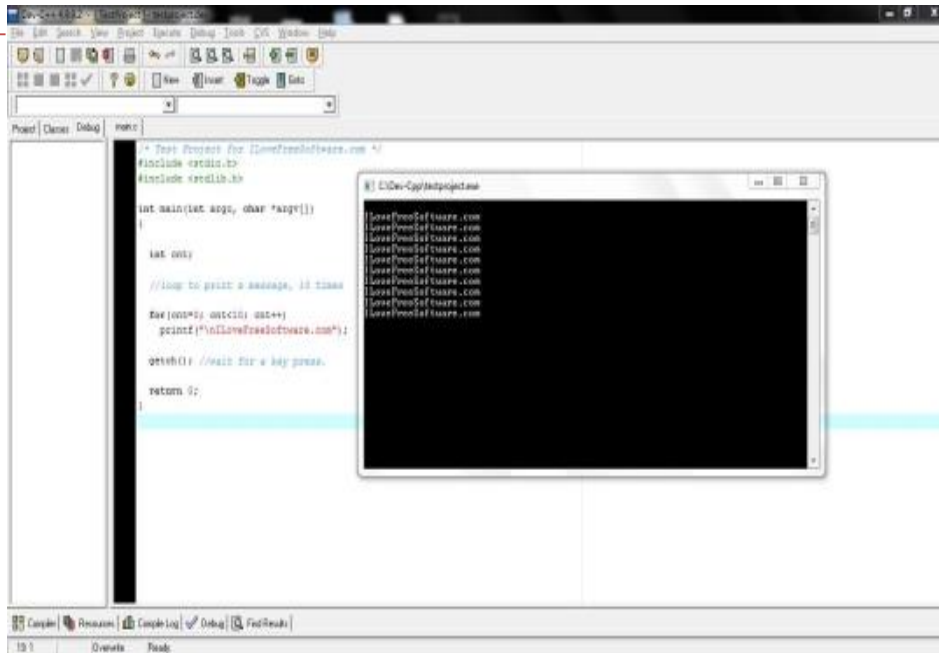
- Un programme est **un ensemble d'instructions élémentaires** écrit dans **un langage de programmation**, et exécuté par un ordinateur afin de traiter les données d'un problème et renvoyer un ou plusieurs résultats

---

- Structure d'un programme en C



# IDE :Integrated Development Environment

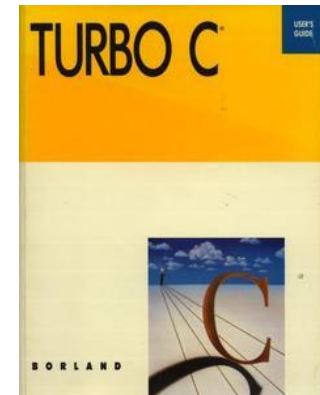
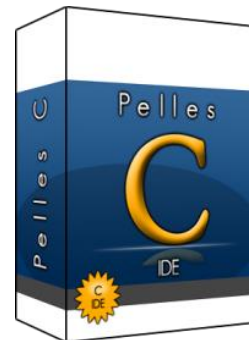


# IDE :Integrated Development Environment

---

**IDE**: *Integrated Development Environment* est un utilitaire ou logiciel de développement qui regroupe :

- **Un Editeur**: Permet la saisie du programme (code source).
- **Un Compilateur**: Traduit le code source du programme en un langage machine.
- **Un débogueur**: Permet de détecter les erreurs du programme.



# Types des données

---

## 1. Entier

Type	Valeur min	Valeur max	Nombre d'octets
Int	- 32768	32767	2
long	-2147483648	2147483647	4
Unsigned int	0	65535	2
Unsigned long	0	4294967295	4



# Types des données

---

## 2. Réel

Type	Valeur min	Valeur max	Nombre d'octets
Float	$3.4 \cdot 10^{-38}$	$3.4 \cdot 10^{+38}$	4
double	$1.7 \cdot 10^{-308}$	$1.7 \cdot 10^{+308}$	8
Long double	$3.4 \cdot 10^{-4932}$	$3.4 \cdot 10^{+4932}$	10





# Types des données

---

## 3. Caractère

Type	Nombre d'octets
char	1



# Types des données

---

## 4. booléen

Type	Valeur	Nombre d'octets
bool	0 ou 1	1



# Déclaration d'une donnée

---

## Syntaxe de déclaration d'une constante

```
Const type Nom_Constante = valeur ;
```

## Syntaxe de déclaration d'une variable

```
Type Nom_Variable ;
```



# Déclaration d'une donnée

---

## Exemples de constantes:

```
Const float pi= 3.14 ;  
Const int nbrs_mois= 12 ;
```

## Exemples de variable

```
int num_etudiant ;  
Float note ;  
Char sexe ;
```



# Affectation

---

## Syntaxe d'affectation :

Nom\_variable  Valeur

## Exemples :

A = 2 ;

la variable A reçoit la valeur 2

B = A

la variable B reçoit le contenu de la variable A

C = A+B

la variable C reçoit le résultat de A+B

D = "M"

la variable D reçoit la valeur M

---

## Exercice : affectation

Ecrire un programme qui permet d'échanger le contenu de deux entier A et B:

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

int main()
{
    int B,A,tmp;
    B=1;
    A=2;
    tmp=A;
    A=B;
    B=tmp;
    printf (" la valeur de A est :%d \n",A);
    printf (" la valeur de B est :%d",B);

    return 0;
}
```



# *Les instructions de base*

## Instruction d'écriture Printf()

### Syntaxe

```
printf ( " Texte .....", variable ) ;
```

- Le texte à afficher
- Les spécificateurs de format
- Caractère d'échappement

Variables , constantes ou  
une expression

# *Les instructions de base*

---

## Instruction d'écriture Printf()

### Exemple

**Printf** ("bonjour " ) ;

Signifie affiché sur l'écran le message : bonjour

**Printf** ( " A = %d", A) ;

Signifie affiché sur l'écran le contenu de la variable A (qui est de type entier).

**Printf** ( " les coordonnées sont : %f ,%f ", X,Y) ;

Signifie affiché sur l'écran le message : les coordonnées sont : plus les contenus de X et Y (qui ont de type float )



# Instruction d'écriture Printf()

---

## Les spécificateur de format

Les spécificateur de format	Type
" %d "	int
" %ld "	long
" %f "	Float ou double
" %c"	Char



# Instruction d'écriture Printf()

---

## Les caractères d'échappement

Les caractères d'échappement	Type
"\n"	Retour à la ligne
"\t"	Tabulation horizontale
"\'"	Affiche une apostrophe
"\""	Affiche un guillemet
"\\"	Affiche un antislash
"%%"	Affiche un %



# *Les instructions de base*

---

## Instruction de lecture scanf()

**Scanf ()** permet de demander à l'utilisateur de fournir des informations  
chaque information donnée par l'utilisateur est **stockée** dans une variable

### Syntaxe

```
scanf( "spécificateur de format" ,& variable ) ;
```

# *Les instructions de base*

---

## Instruction de lecture Scanf()

### Exemple

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

Lire la valeur de X (X est un entier )

```
scanf ("%c", &Y);
```

Lire la valeur de Y (Y est un caractère )

```
scanf ("%d%d%d", &jour, &mois, &annee);
```

Pour lire la valeur du jour du mois et d'année à la fois

**Remarque :** il faut pas mettre les virgules entre les spécificateurs de format

# Exercice : affectation

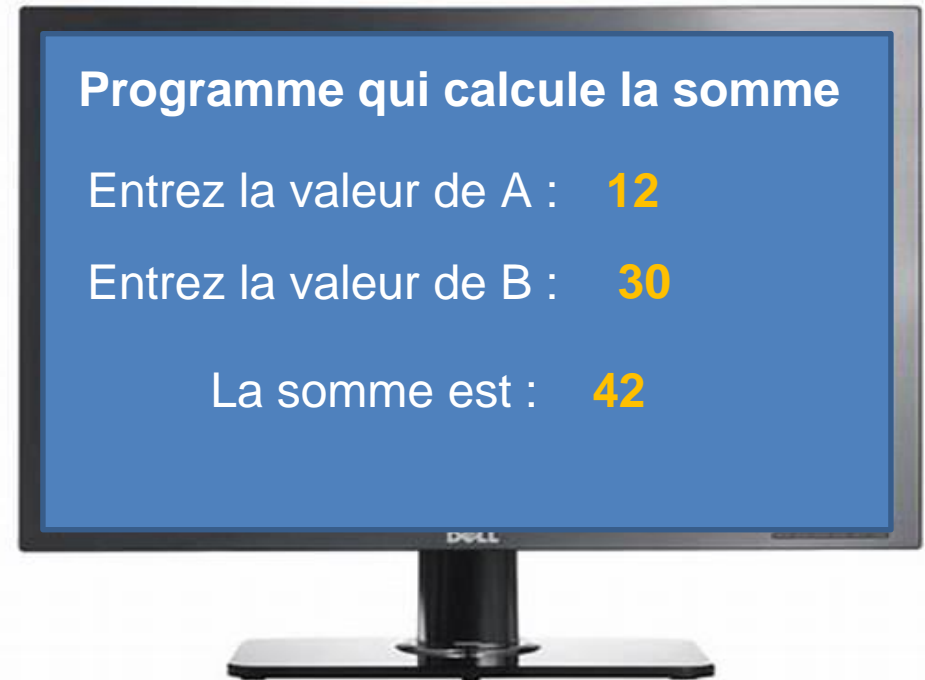
Ecrire un programme qui permet :

- 1- Demander à l'utilisateur d'entrer deux entier A et B .
- 2- d'échanger le contenu de A et B.
- 3- Afficher le résultat .

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int B,A,tmp;
    printf("entrez la valeur de A \t");
    scanf("%d",&A);
    printf("entrez la valeur de B \t");
    scanf("%d",&B);
    tmp=A;
    A=B;
    B=tmp;
    printf (" la valeur de A est :%d \n",A);
    printf (" la valeur de B est :%d",B);

    return 0;
}
```

# *Les instructions de base*



## Exercice : la somme de trois entiers

---

Ecrire un programme qui permet de calculer la somme de trois entiers

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int B,A,C,som;
    printf("entrez le premier entier \t");
    scanf("%d",&A);
    printf("entrez le deuxième entier \t");
    scanf("%d",&B);
    printf("entrez le troisième entier \t");
    scanf("%d",&C);
    som=A+B+C;

    printf (" la somme est : %d",som);

    return 0;
}
```



# *Les instructions de base*

## Exercice :

Écrire un programme qui calcule la surface d'un cercle

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

int main()
{
    const float pi=3.14;
    float rayon,surface ;
    printf("programme qui calcule la surface d'un cercle \n");
    printf("entrez un rayon \t");
    scanf("%f",&rayon);

    surface=(rayon*rayon*pi);

    printf (" la surface est : %f\t",surface);
    return 0;
}
```





## Exercice :

Ecrire un programme qui permet :

- 1- Demander à l'utilisateur d'entrer deux entier A et B .
- 2- d'incrémenter A par 4.
- 3- Afficher la somme de A+B.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a,b,som;
    printf("entrez le premier nombre \t");
    scanf("%d",&a);
    printf("entrez le deuxième nombre \t");
    scanf("%d",&b);
    a-=4;
    som=a+b;
    printf ("la somme est :%d\t",som);
    return 0;
}
```

## Exercice :

Ecrire un programme qui permet calculer la somme ,le produit et la soustraction de deux nombres entier.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a,b,prod,sous,som;
    printf("entrez le premier nombre \t");
    scanf("%d",&a);
    printf("entrez le deuxième nombre \t");
    scanf("%d",&b);
    prod=a*b;
    sous=a-b;
    som=a+b;
    printf ("la somme est :%d \n",som);
    printf ("le produit est :%d\n",prod);
    printf ("la soustraction est :%d",sous);
    return 0;
}
```

## Exercice :

Ecrire un programme qui demande à l'utilisateur la saisie des valeurs de deux nombres X et Y ,puis calcule et affiche X à la puissance Y.

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

int main()
{
    int x,y,res;
    printf("entrez le premier nombre \t");
    scanf("%d",&x);
    printf("entrez le deuxième nombre \t");
    scanf("%d",&y);
    res=pow(x,y);
    printf ("%d a la puissance %d = %d",x,y,res);
    return 0;
}
```

## Exercice :

Ecrire un programme qui demande à l'utilisateur d'entrer 3 notes et puis le programme affiche la moyenne .

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    float not1,not2,not3,moy;
    printf("entrez le premier nombre \t");
    scanf("%f",&not1);
    printf("entrez le deuxième nombre \t");
    scanf("%f",&not2);
    printf("entrez le deuxième nombre \t");
    scanf("%f",&not3);
    moy=(not1+not2+not3)/3;
    printf ("%f",moy);
    return 0;
}
```

## Exercice :

Ecrire un programme qui demande à l'utilisateur d'entrer le rayon d'une sphère, puis calcule et affiche son volume.

$$\text{Volume} = (4 * \pi * r^3) / 3$$

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    const float pi=3.14;
    float rayon,volume ;

    printf("entrez un rayon \t");
    scanf("%f",&rayon);

    volume=(4*pi*pow(rayon,3))/3;

    printf (" la surface est : %f\t",volume);
    return 0;
}
```



## Exercice :

Ecrire un programme qui demande une durée(entier) exprimé en secondes, et qui convertit en heures,minutes,secondes.

Exemple :

**duree= 67890 secondes ⇔ 18 heures 51 minutes 30 secondes**

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

int main()
{
    int duree,H,Reste,M,S;
    printf("entrez une durée en seconde");
    scanf("%d",&duree);
    H=duree/3600;
    Reste=duree%3600;
    M=Reste/60;
    S= Reste%60;
    printf("%d heures %d minutes %d secondes",H,M,S);
    return 0;
}
```

## Exercice :

---

Ecrire un programme qui calcule et affiche la distance entre deux points A et B du plan dont les coordonnées  $(X_a, Y_a)$  et  $(X_b, Y_b)$  sont entrées au claviers comme entiers.

**Remarque :** utiliser les fonctions `pow()` et `sqrt()` pour calculer la puissance et la racine carrée.



## Exercice :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int xa,ya,xb,yb;
    float AB;
    printf("entrer la valeur de xa");
    scanf("%d",&xa);
    printf("entrer la valeur de ya");
    scanf("%d",&ya);
    printf("entrer la valeur de xb");
    scanf("%d",&xb);
    printf("entrer la valeur de yb");
    scanf("%d",&yb);
    AB=sqrt( pow(xb-xa,2)+pow(yb-ya,2));
    printf("la distance entre A et B est : %f",AB);
    return 0;
}
```





# Expression arithmétique

---

Les opérateurs arithmétiques usuels sont :

Opérateur	signification
+ , -	Addition , Soustraction
* - /	Multiplication , Division
%	Reste de la division entière
++	Incrémentation
--	décrémentation

$A=B++$ ; On affecte B à A et puis on fait  $B=B+1$ .

$A=++B$ ; On fait  $B=B+1$  puis on affecte la nouvelle valeur de B à A.



# Expression arithmétique élargie

---

Exemple des opérateurs arithmétiques élargies:

Opérateur	Opération normale	Opération élargie
<code>+=</code>	<code>X=X+Y;</code>	<code>X+=Y;</code>
<code>-=</code>	<code>X=X-Y;</code>	<code>X-=Y</code>
<code>*=</code>	<code>X=X*Y;</code>	<code>X*=Y</code>
<code>/=</code>	<code>X=X/Y;</code>	<code>X/=Y;</code>
<code>%=</code>	<code>X=X%Y;</code>	<code>X%=Y;</code>



# Expression arithmétique élargie

---

Exemple des opérateurs arithmétiques élargies:

Instruction	A	B	C	D
A=9;				
B=++A;				
C=A*B++;				
D=pow(C,2);				
A/=2;				
D=++D-C;				
C-=B--;				
B=(A--++-C)/3;				



# Expression arithmétique élargie

Exemple des opérateurs arithmétiques élargies:

Instruction	A	B	C	D
A=9;	9			
B=++A;	10	10		
C=A*B++;	10	11	100	
D=pow(C,2);	10	11	100	10000
A/=2;	5	11	100	10000
D=++D-C;	5	11	100	9901
C-=B--;	5	10	89	9901
B=(A--++-C)/3;	4	31	88	9901



# Expression de comparaison

---

Une expression de comparaison donne un résultat **booléen** . Les opérateurs de comparaison usuels sont:  $>$  ,  $==$  ,  $<$  ,  $>=$  ,  $<=$  ,  $!=$  .



# Expression logique

---

ET  $\Leftrightarrow$  &&

Ou  $\Leftrightarrow$  ||

Non  $\Leftrightarrow$  !



# *Les structure conditionnelle*

---

*La structure conditionnelle simple*

*Structure conditionnelle alternative (deux choix)*

*Structure alternative imbriquée (multiple choix)*

*Structure conditionnelle à choix multiple*



# Structure conditionnelle

---

## 1. Structure conditionnelle Simple (un choix)

Syntaxe :

```
if (Condition )  
{  
    instruction 1;  
    instruction 2;  
    .....  
    instruction n  
}
```

### Remarque

Si on a une seule instruction c'est pas la peine d'ajouter les accolades





# *Structure conditionnelle*

---

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int A,B;
    float division;
    printf("Svp entrez le dividende ");
    scanf("%d",&A);
    printf("Svp entrez le diviseur");
    scanf("%d",&B);
    if(B!=0){
        division=(float)A/B;
        printf("le resultat de le division est :%f",division);
    }
    return 0;
}
```



# *Structure conditionnelle*

Ecrire un programme qui permet de calculer le maximum de deux nombres entiers saisis par l'utilisateur.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A,B,max;

    printf("Svp entrez le premier nombre");
    scanf("%d",&A);
    printf("Svp entrez le deuxieme nombre ");
    scanf("%d",&B);
    max=A;
    if(A<B)
        max=B;
    printf("le max est :%d ",max);
    return 0;
}
```



# *Structure conditionnelle alternative*

---

## 2. Structure conditionnelle alternative (deux choix)

Syntaxe :

```
if (Condition) {  
    instruction 1;  
    .....  
    instruction n;  
}  
else {  
    instructions 2;  
    .....  
}
```



# *Structure conditionnelle*

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A,B;
    float division;
    printf("Svp entrez le dividende ");
    scanf("%d",&A);
    printf("Svp entrez le diviseur");
    scanf("%d",&B);
    if(B!=0){
        division=(float)A/B;
        printf("le resultat de le division est :%f",division);
    }
    else
        printf("la division par zero est impossible");
    return 0;
}
```



# ***Structure conditionnelle alternative***

Ecrire un programme qui permet de calculer le maximum de deux nombres entiers saisis par l'utilisateur.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A,B,max ;
    printf("Svp entrez le premier nombre ");
    scanf("%d",&A);
    printf("Svp entrez le deuxieme nombre ");
    scanf("%d",&B);
    if(A<B)
        max=B;
    else
        max=A;
    printf ("le maximum est %d:" , max );
    return 0;
}
```



# ***Structure conditionnelle alternative***

Ecrire un programme qui permet de demander un nombre entier à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif .

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int N ;
    printf("Svp entrez le premier nombre ");
    scanf("%d",&N);

    if(N>0)
        printf ("le nombre est positif" );
    else
        printf ("le nombre est negatif" );
    return 0;
}
```



# Structure alternative imbriquée

---

## 3. Structure alternative imbriquée (multiple choix)

Syntaxe :

```
if ( Condition1 ) {  
    instructions 1  
}  
else if ( Condition2 ) {  
    instructions 2  
}  
else if ( Condition3 ) {  
    .....  
}  
else {  
    instructions 3  
}
```



# *Structure alternative imbriquée*

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int N ;
    printf("Svp entrez le premier nombre ");
    scanf("%d",&N);
    if(N>0)
        printf ("le nombre est positif" );
    else if (N==0)
        printf ("le nombre est Null" );
    else
        printf ("le nombre est negatif" );
    return 0;
}
```





# *Structure alternative imbriquée*

---

## **Exercice**

Ecrire un programme qui permet de lire la valeur de la température de l'eau et d'afficher son état :

- **Glace** si la température est inférieure à 0
- **Liquide** si la température strictement supérieure à 0 et strictement inférieure à 100
- **Vapeur** si la température est supérieure à 100



# *Structure alternative imbriquée*

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int T ;
    printf("Svp entrez la température de l'eau :");
    scanf("%d",&T);
    if(T<=0)
        printf ("Glace" );
    else if (T<100)
        printf ("liquide" );
    else
        printf ("vapeur" );
    return 0;
}
```



# Structure conditionnelle à choix multiple

---

## 4. Structure conditionnelle à choix multiple

Syntaxe :

```
switch ( Variable ){  
    case Valeur1 : instructions 1;  
                    break;  
    case Valeur2 : instructions 2;  
                    break;  
  
    .....  
    case ValeurN : instructions N;  
                    break;  
  
    default : autre instruction ;  
              break;  
  
}
```



# Structure conditionnelle à choix multiple

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int j ;
    printf("veuillez entrer un nombre entre 1 et 7 :");
    scanf("%d",&j);
    switch(j){
        case 1: printf("lundi");
            break;
        case 2: printf("mardi");
            break;
        case 3: printf("mercredi");
            break;
        case 4: printf("jeudi");
            break;
        case 5: printf("vendredi");
            break;
        case 6: printf("samedi");
        case 7: printf("dimanche");
            break;
        default : printf("num incorrect");
            break;
    }
    return 0;
}
```



---

TD



## Exercice

---

Ecrire un programme **minimum** qui demande à l'utilisateur de saisir trois nombre et qui affiche leur minimum

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A,B,C,min ;
    printf("veuillez entrer 3 nombres:");
    scanf("%d%d%d",&A,&B,&C);
    if(A<B && A<C)
        printf("le min est :%d",A);
    else if (B<C)
        printf("le min est :%d",B);
    else
        printf("le min est :%d",C);

    return 0;
}
```



---

## Exercice

Ecrire un programme **signe** qui retourne si deux nombres entiers donnés sont de même signe ou non.

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

int main()
{
    int A,B;
    printf("veuillez entrer 2 nombres:");
    scanf("%d%d",&A,&B);
    if((A*B >0)){
        printf("meme signe");
    }
    else{
        printf("deux signes différents");
    }
    return 0;
}
```




---

## Exercice

Ecrire un programme **PaireImpaire** qui demande un nombre à l'utilisateur et l'informe ensuite si ce nombre est pair ou impair

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A;
    printf("veuillez entrer un nombre:");
    scanf("%d",&A);
    if( A%2 == 0 ){
        printf("le nombre est Paire");
    }
    else{
        printf("le nombre est imPaire");
    }
    return 0;
}
```





## Exercice

Ecrire un programme **Echange\_cond** qui échange les contenus de deux nombres entiers donnés si elles sont de même signe sinon il met la somme des deux dans la première et leur produit dans la seconde .

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A,B,tmp,som,prod;
    printf("veuillez entrer 2 nombres:");
    scanf("%d%d",&A,&B);
    if(A*B>0){
        tmp=A;
        A =B;
        B =tmp;
    }
    else{
        som=A+B;
        prod=A*B;
        A=som;
        B=prod; }
    printf("la valeur de A est : %d et la valeur de b est : %d",A,B);
    return 0;
}
```

## Exercice

---

Ecrire un programme qui permet de résoudre **une équation du second degré**.

$$ax^2+bx+c=0$$

$$\Delta = b^2 - 4ac.$$

- Si  $\Delta > 0$ , alors l'équation admet deux solutions réelles notées  $x_1$  et  $x_2$ .

On a alors :

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a} \text{ et } x_2 = \frac{-b + \sqrt{\Delta}}{2a}.$$

- Si  $\Delta = 0$ , alors l'équation admet une solution réelle double notée  $x_0$ ;

$$\text{on a alors : } x_0 = \frac{-b}{2a} ;$$

- Si  $\Delta < 0$ , Pas de solution

---



---

## Etude du trinôme : $x^2 + 2x + 1$

On identifie les valeurs des coefficients :  $a = 1$  ;  $b = 2$  ;  $c = 1$

**Discriminant du trinôme :**  $\Delta = b^2 - 4ac = 0$

Racine du discriminant :  $\sqrt{\Delta} = 0$

**Le discriminant  $\Delta$  est nul, l'équation  $x^2 + 2x + 1 = 0$  admet une solution :**  $x_1 = \frac{-b}{2a} = -1$



---

**Etude du trinôme :  $2x^2 - 9x - 5$**

**Discriminant du trinôme :  $\Delta = b^2 - 4ac = 121$**

Racine du discriminant :  $\sqrt{\Delta} = 11$

**Le discriminant  $\Delta$  est strictement positif, l'équation  $2x^2 - 9x - 5 = 0$  admet deux solutions.**

$$\text{Solution 1 : } x_1 = \frac{-b - \sqrt{\Delta}}{2a} = \frac{9 - 11}{4} = \frac{-1}{2}$$

$$\text{Solution 2 : } x_2 = \frac{-b + \sqrt{\Delta}}{2a} = \frac{9 + 11}{4} = 5$$



---

### Etude du trinôme : $3x^2 + 2x + 6$

On identifie les valeurs des coefficients :  $a = 3$  ;  $b = 2$  ;  $c = 6$

Discriminant du trinôme :  $\Delta = b^2 - 4ac = -68$

Le discriminant  $\Delta$  est strictement négatif, l'équation  $3x^2 + 2x + 6 = 0$  n'admet aucune solution réelle.



```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int a,b,c,delta;
    float x1,x2,x0;
    printf("donnez la valeur de a");
    scanf("%d",&a);
    printf("donnez la valeur de b");
    scanf("%d",&b);
    printf("donnez la valeur de c");
    scanf("%d",&c);
    delta=pow(b,2)-4*a*c;
```

```
if(delta>0){
    x1=(-b-sqrt(delta))/(2*a);
    x2=(-b+sqrt(delta))/(2*a);
    printf("les deux solutions sont :\n x1= %f \n
x2= %f",x1,x2);
}
else if(delta==0){
    x0=(-b)/(2*a);
    printf("la solution est : x0= %f",x0);
}
else{
    printf("pas de solution");
}
return 0;
}
```



## Exercice

---

Ecrire un programme, qui à partir du numéro d'un mois saisi pas l'utilisateur, permet d'afficher la saison .

Si mois = 6,7 ou 8 alors le programme affiche Eté.

Si mois = 1, 2 ou 12 le programme affiche Hiver .

Si mois = 3,4 ou 5 le programme affiche printemps .

Si mois = 9,10 ou 11 le programme affiche automne



```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int mois;
  printf("donnez le numero d'un mois");
  scanf("%d",&mois);
  if(mois<=2){
    printf("hiver");
  }else if(mois<=5){
    printf("printemps");
  }else if(mois<=8){
    printf("ete");
  }
  else if(mois<=11){
    printf("automne");
  }
  else{
    printf("hiver");
  }
  return 0;
}
```

---





# Exercice

Un magasin de photocopie facture **0,30 dh** les dix premières photocopies , **0,25 dh** les vingt suivantes et **0,20 dh** au-delà. Ecrire un programme qui demande à l'utilisateur le nombre de copies effectuées et qui affiche la facture correspondante.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int nbrC;
  float F;
  printf("donnez le nombre de copies");
  scanf("%d",&nbrC);
  if(nbrC<=10){
    F=nbrC*0.30;
  }
  else if(nbrC<=30) {
    F=(10*0.30)+((nbrC-10)*0.25);
  }
  else{
    F=(10*0.30)+(20*0.25)+((nbrC-30)*0.20);
  }
  printf("le montant à payer est :%f",F);
  return 0;
}
```

# Exercice

---

Ecrire un programme permettant de saisir trois note (sur 20) d'un étudiant, calculant sa moyenne et affichant cette moyenne avec la mention ( "très bien " à partir de 16, "bien" entre 14 et 16 , "Assez bien " entre 12 et 14, "Passable" entre 10 et 12 , "insuffisant" en dessous de 10 )

**Rmq :** on suppose que l'étudiant va saisir des notes compris entre 0 et 20.



```
#include <stdio.h>
#include <stdlib.h>
int main()
{
float n1,n2,n3,moy;
    printf("donnez trois notes");
    scanf("%f%f%f",&n1,&n2,&n3);
    moy=(n1+n2+n3)/3;

    if(moy>=16){
        printf("trés bien");
    }
    else if(moy>=14) {
        printf("bien");
    }
    else if(moy>=12){
        printf("assez bien ");
    }
    else if(moy>=10){
        printf("passable");
    }
    else{
        printf("insuffisant");
    }
    return 0;
}
```



# *Structure répétitives*

---

**La structure répétitive (Boucle)** permet d'exécuter plusieurs fois une séquence d'instructions.

Dans une boucle, **le nombre de répétitions** peut être connu , **fixé à l'avance** , comme il peut **dépendre d'une condition** permettant l'arrêt et la sortie de cette boucle



# Structure répétitives

---

Boucle for



Le nombre de  
répétitions peut être  
connu

Boucle while



Boucle do ..... while



Le nombre de  
répétitions dépend  
d'une condition



# *Structure répétitives « for »*

---

**Boucle for**



Le nombre de  
répétitions peut être  
connu



# Structure répétitives « for »

---

## 1. La boucle « for »

Cette boucle permet d'exécuter une séquence d'instructions un nombre de fois connu fixé à l'avance.

Syntaxe :

***for ( initialisation ; condition ; incrémentation ) {***

***instructions***

***}***



# *Structure répétitives « for »*

---

## 1. La boucle « for »

**initialisation:** est une instruction exécutée avant le premier tour de la boucle .

**Condition :** tant que cette condition est vraie, la boucle for continue.

**incrémentation :** cette instruction est exécutée à la fin de chaque tour de boucle pour mettre à jour la variable.





# Structure répétitives « for »

---

## 1. La boucle « for »

**initialisation:** la boucle suivante affiche le message « bonjour DSI » 3 fois.

```
int main()
{
    int i ;
    for (i=0 ; i<3 ; i++)
    {
        printf("bonjour DSI \n");
    }
    return 0;
}
```



# Structure répétitives « for »

---

## Table de multiplication de 7

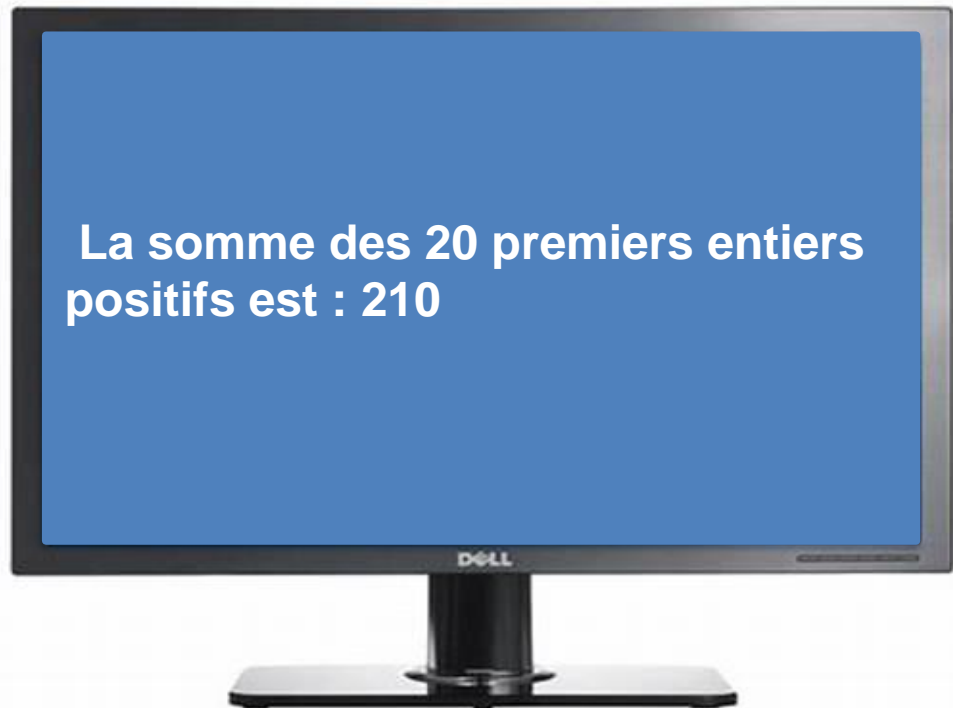
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i,m;
    printf("la table de multiplication de 7 \n");
    for(i=0;i<=10;i++){
        m=i*7;
        printf("%d * 7 =%d \n",i, m);
    }
    return 0;
}
```



# Exercice : la boucle for

---

Ecrire un programme qui permet de calculer la somme des 20 premiers entiers positifs.



# Exercice : la boucle for

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i,som;
    som=0;
    for(i=1;i<=20;i++){
        som=som+i;
    }
    printf("la somme des 20 premiers entier positif est %d ",som);
    return 0;
}
```



# ***Structure répétitives***

---

Boucle while



Le nombre de  
répétitions dépend  
d'une condition



# Structure répétitives « while »

---

## 2. La boucle « tant que »

Cette boucle permet de répéter un bloc d'instructions **tant qu'**une **condition** est **vraie**

Syntaxe :

```
while (Condition) {  
    instructions  
}
```



# Structure répétitives « while »



## Table de multiplication

Veillez saisir un nombre entre 1 et 10 : -23

Veillez saisir un nombre entre 1 et 10 : 14

Veillez saisir un nombre entre 1 et 10 : 3

$$0 * 3 = 0$$

$$1 * 3 = 3$$

$$2 * 3 = 6$$

$$3 * 3 = 9$$

$$4 * 3 = 12$$

$$5 * 3 = 15$$

$$6 * 3 = 18$$

$$7 * 3 = 21$$

$$8 * 3 = 24$$

$$9 * 3 = 27$$

$$10 * 3 = 30$$

# Structure répétitives « while »

## Table de multiplication

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
while(n<1 || n>10){
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
}
for(i=0;i<=10;i++){
    m=i*n;
    printf("%d * %d =%d \n",i,n, m);
}

return 0;
}
```





# Structure répétitives « while »

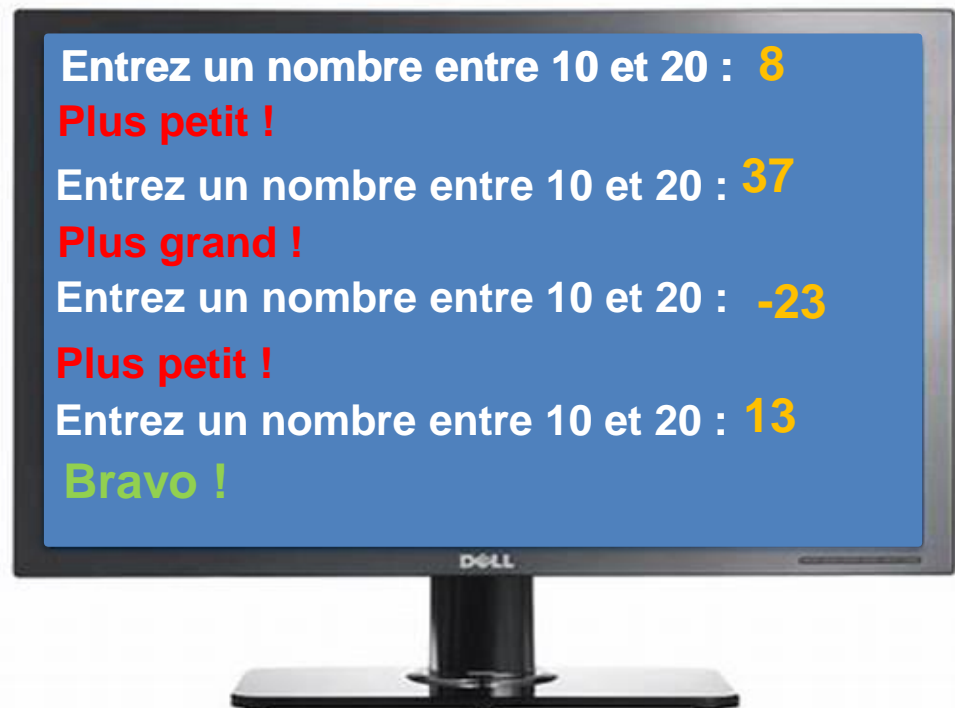
## Table de multiplication

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
while(n<1 || n>10){
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
}
i=0;
while (i<=10){
    m=i*n;
    printf("%d * %d =%d \n",i,n, m);
    i++;
}

return 0;
}
```

# Exercice : la boucle while

Ecrire un programme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de la réponse supérieure à 20 on fera apparaître un message : « plus grand ! » et inversement, « plus petit ! » si le nombre est inférieur à 10



# *Structure répétitives « while »*

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;
printf("Veuillez saisir un nombre entre 10 et 20 :");
scanf("%d",&n);
while(n<10 || n>20){
    if(n<10){
        printf("plus petit \n");
    }
    else{
        printf("plus grand \n");
    }
printf("Veuillez saisir un nombre entre 10 et 20 :");
scanf("%d",&n);
}
printf("bravo");

    return 0;
}
```



# ***Structure répétitives***

---

Boucle do ..... while



Le nombre de  
répétitions dépend  
d'une condition



# Structure répétitives « *do... while* »

---

## 2. La boucle « *do ... while ...*»

Cette boucle permet de répéter un bloc d'instructions tant qu'une condition est vraie.

Syntaxe :

```
do {  
    instructions;  
} while (Condition) ;
```



# Structure répétitives « do ... while .. »

---

## Table de multiplication

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;

do{
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
}while(n<1 || n>10);
for(i=0;i<=10;i++){
    m=i*n;
    printf("%d * %d =%d \n",i,n, m);
}
return 0;
}
```



---

## La boucle do..... while ..

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;

do{
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
}while(n<1 || n>10);

for(i=0;i<=10;i++){
    m=i*n;
    printf("%d * %d =%d \n",i,n, m);
}

return 0;
}
```

## La boucle while

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,m;

printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);

while(n<1 || n>10){
printf("Veuillez saisir un nombre entre 1 et 10 :");
scanf("%d",&n);
}

for(i=0;i<=10;i++){
    m=i*n;
    printf("%d * %d =%d \n",i,n, m);
}

return 0;
}
```



## Exercice : la boucle do.. while

---

Ecrire un programme qui demande à l'utilisateur de saisir un nombre entier strictement supérieur à 1 , et qui calcule la somme des entiers jusqu'à ce nombre .

Par exemple si l'on entre 5, le programme doit calculer :  $1+2+3+4+5$





# Structure répétitives « do ... while »

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n,i,som;

do{
printf("Veuillez saisir un nombre strictement
supérieur a 1 :");
scanf("%d",&n);
}while(n<=1);
som=0;
for(i=1;i<=n;i++){
    som+=i;
}
printf("la somme est :%d",som);
return 0;
}
```



---

**TD**



---

### Exercice 3 :

Ecrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. par exemple, si l'utilisateur entre le nombre 34, le programme affichera les nombres de 35 à 44.



## ***Exercice 3***

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int nbr,i;
printf("donnez un nombre de départ ");
scanf("%d",&nbr);
for(i=nbr+1;i<=nbr+10;i++){
    printf("%d\n",i);
}
return 0;
}
```



---

### Exercice 3 :

Ecrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants(en utilisant la boucle **while** et **do .. While** ).par exemple, si l'utilisateur entre le nombre 34, le programme affichera les nombres de 35 à 44.



## Exercice 3

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int nbr,i;
printf("donnez un nombre de départ ");
scanf("%d",&nbr);
i=nbr+1;
while(i<=nbr+10){
printf("%d\n",i);
i++;
}
return 0;
}
```



## Exercice 3

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int nbr,i;
printf("donnez un nombre de départ ");
scanf("%d",&nbr);
i=nbr+1;
do{
printf("%d\n",i);
i++;
}while(i<=nbr+10);
return 0;
}
```



---

### Exercice 3 :

Ecrire un programme qui affiche les diviseurs d'un entier positif  $n$  non nul .






## Exercice 3

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int n,i;
    printf(" entrez un nombre positif non null :");
    scanf("%d",&n);
    if(n>0){
        for(i=1;i<=n;i++){
            if(n%i==0){
                printf("%d est le diviseur de %d \n",i,n);
            }
        }
    }
    else{
        printf("entrez un nombre positif non null");
    }

    return 0;
}
```



---

## Exercice 4 :

Ecrire un algorithme qui calcule et affiche la somme :

$$S = 1/1 + 1/2 + \dots + 1/n$$



## ***Exercice 4***

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int n,i;
float som,v;
    printf(" entrez un nombre ");
    scanf("%d",&n);
    som=0;
    for(i=1;i<=n;i++)
    {
        som=som+(float)1/i;
    }
    printf("la somme est:%f",som);
    return 0;
}
```



---

## Exercice 5 :

Ecrire un algorithme qui calcule et affiche la somme :

$$S = 10^0 + 10^1 + 10^2 + 10^3 \dots\dots + 10^n$$



## ***Exercice 5***

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int n,i,som;
  printf(" entrez un nombre ");
  scanf("%d",&n);
  som=0;
  for(i=0;i<=n;i++)
  {
    som+=pow(10,i);
  }
  printf("la somme est:%d",som);
  return 0;
}
```



## ***Exercice 6***

---

### **Exercice 6 :**

Ecrire un programme qui demande à l'utilisateur de taper un entier  $n$  , puis qui calcule le somme des carrées des  $n$  premiers entier impairs par exemple ,  
si  $n=5$  le résultats est :  **$1^2 + 3^2 + 5^2 + 7^2 + 9^2 = 165$**



## Exercice 6

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int n,i,j,som;
  printf(" entrez un nombre ");
  scanf("%d",&n);
  som=0;
  j=1;
  for(i=1;i<=n;i++)
  {
    som+=pow(j,2);
    j=j+2;
  }
  printf("la somme est:%d",som);
  return 0;
}
```

N	5	i		S		j	
N	5	i		S	0	j	1
N	5	i	1	S	1	j	3
N	5	i	2	S	10	j	5
N	5	i	3	S	35	j	7
N	5	i	4	S	84	j	9
N	5	i	5	S	165	j	11
N	5	i	6	S	165	j	11



## ***Exercice 6***

---

### **Exercice 6 :**

Ecrire un programme qui demande à l'utilisateur de saisir son âge (un entier positif) puis l'algorithme lui affiche son stade de vie:

- Bébé : de la naissance à 2 ans .
- Enfant : de 3ans à 12 ans .
- Adolescent : de 13 ans à 17ans
- Adulte : de 18ans à 69ans
- Personne âgée : à partir de 70 ans .
- Effectuer un contrôle de saisie sur l'âge qui doit être positif .





## Exercice 6

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int age;
  printf(" donnez votre age ");
  scanf("%d",&age);
  while(age<0){
    printf("l'age doit être positif ! veuillez
saisir un age positif: ");
    scanf("%d",&age);
  }
```

```
    if(age<=2)
      printf("bebe");
    else if(age<= 12)
      printf("enfant");
    else if(age <= 17 )
      printf("adolescent");
    else if(age <= 69 )
      printf("adulte");
    else
      printf("Personne agee");
    return 0;
  }
```



---

## Exercice 1 :

La population de la ville Marrakech est de **1 000 000** d'habitants et elle augmente de **50 000** habitants par an. Celle de la ville Agadir est de **500 000** habitants et elle augmente de **8%** par an . Ecrire un algorithme permettant de déterminer dans combien d'années la population de la ville Agadir dépassera celle de la ville Marrakech.



# Exercice 1

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int nbr_Agadir,nbr_Kech,nbr_Ans;
  nbr_Agadir=500000 ;
  nbr_Kech=1000000 ;
  nbr_Ans=0 ;
  while(nbr_Agadir < nbr_Kech){

    nbr_Agadir=nbr_Agadir + (nbr_Agadir * 8/100) ;
    nbr_Kech=nbr_Kech + 50000 ;
    nbr_Ans=nbr_Ans + 1 ;
  }
  printf("Agadir dépassera marakechh après %d ans",nbr_Ans);
  return 0;
}
```



---

## Exercice 2 :

A la naissance de Amal , son grand-père Ali , lui ouvre un compte bancaire. Ensuite ,à chaque anniversaire, le grand père de Amal verse sur son compte 500dh , auxquels il ajoute le triple de l'âge de Amal. Par exemple, lorsqu'elle a quatre ans il lui verse 512dh . Ecrire un programme qui permet de déterminer quelle somme aura Amal lors de son nième anniversaire.



## Exercise 2

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{int i,age,s;
  printf("entrez l'age de Amal :");
  scanf("%d",&age);
  s=0;
  for(i=1;i<=age;i++){
    s=s+(500+(i*3)) ;
  }
  printf("le compte d'Amal au %d ième anniversaire est %d:",age, s );
  return 0;
}
```



---

## Exercice :

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

C'était le nombre numéro 2



## ***Exercice 2***

---

```
int main()
{int i,n,max,indice;
max=0;
  for(i=1;i<=6;i++){
    printf("Entrez le nombre numéro %d\t",i);
    scanf("%d",&n);
    if(n>max){
      max=n;
      indice=i;
    }
  }
  printf("le max est %d ",max);
  printf("C'était le nombre numero %d ",indice);
  return 0;
}
```



**Stocker les notes  
des étudiants**

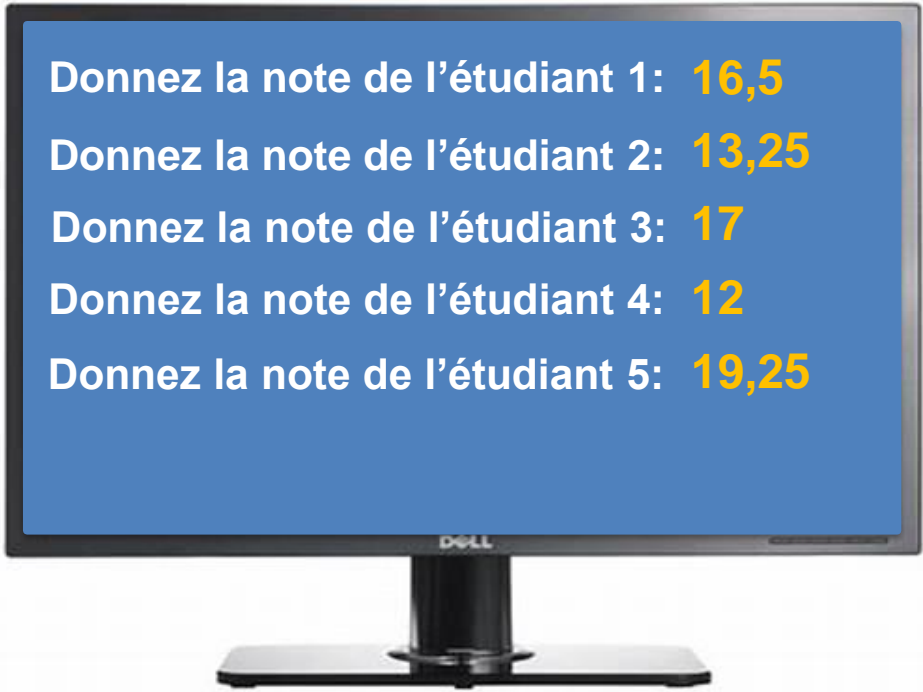


Etudiant	Note
ahmed	19,25
Khalid	15,5
hiba	13,75
Malak	12



---

**stocker**



Donnez la note de l'étudiant 1: **16,5**  
Donnez la note de l'étudiant 2: **13,25**  
Donnez la note de l'étudiant 3: **17**  
Donnez la note de l'étudiant 4: **12**  
Donnez la note de l'étudiant 5: **19,25**

A Dell monitor with a black frame and stand. The screen is blue and displays the text in white and yellow. The Dell logo is visible on the bottom bezel.

---

**Algorithme** note

**Variables** N1,N2,N3,N4 :Reel

**Début**

**Ecrire**(' Donner la note de l'éducation num 1:');  
**Lire**(N1);

**Lire**(N1);

**Ecrire**(' Donner la note de l'éducation num 2:');  
**Lire**(N2);

**Lire**(N2);

**Ecrire**(' Donner la note de l'éducation num 3:');  
**Lire**(N3);

**Lire**(N3);

**Ecrire**(' Donner la note de l'éducation num 4:');  
**Lire**(N4);

**Lire**(N4);

**Fin**



---

**Algorithme** note

**Variables** N1,N2,N3,.....N1000 :Reel

**Début**

**Ecrire**(" Donner la note de l'éducation num 1:");

**Lire**(N1);

**Ecrire**(" Donner la note de l'éducation num 2:");

**Lire**(N2);

**Ecrire**(" Donner la note de l'éducation num 3:");

**Lire**(N3);

.....

**Ecrire**(" Donner la note de l'éducation num 1000:");

**Lire**(N1000);

**Fin**



# *Tableaux*

---

## Un tableau

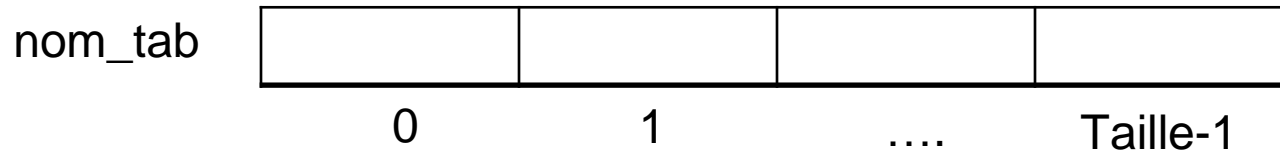
Un tableau est une suites d'éléments de même type. il utilise plusieurs cases mémoire à l'aide d'un seul nom. Comme toutes les cases portent le même nom . Elles se différencient par un indice .



# Déclaration d'un tableau

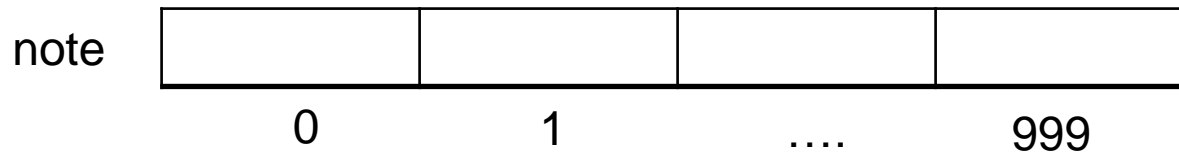
---

**Syntaxe :**    **Tableau** nom\_tab (**taille**) : **type**



Exemple:

**Tableau** note (**1000**) : **Réel**



# Accéder aux éléments d'un tableau

---

**Syntaxe d'affectation:**    Nom\_tab (indice) ← valeur

**Syntaxe de lecture :**        Lire ( Nom\_tab (indice) )

**Syntaxe d'écriture :**        Ecrire ( Nom\_tab (indice) )

## Exemple:

**note** (0) ← 15                      Affectation de la note 15 à l'étudiant num 1

**Lire** (**note** (2) )                      Saisir la note de l'étudiant num 3

**Ecrire** (**note** (6) )                      Affichage de la note de l'étudiant num 7



# Tableaux

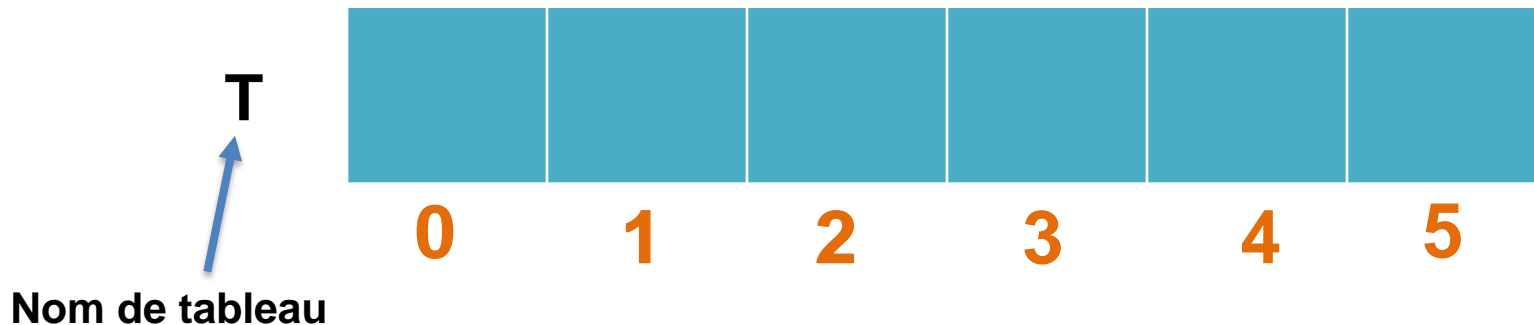
---

## Exemple:

Déclaration d'un tableau nommé T composé de 6 éléments entier:

Tableau T (6) : entier

Le tableau est représenté schématiquement dans la mémoire comme suit:



# Tableaux

## Remplir un tableau

Syntaxe d'affectation :  $\text{Nom\_tab (indice)} \leftarrow \text{valeur}$

Exemple:

$T(0) \leftarrow 3$

$T(1) \leftarrow T(0) + 1$

$T(2) \leftarrow T(0) * T(1)$

$T(3) \leftarrow 15$

$T(4) \leftarrow T(1) - T(0)$

$T(5) \leftarrow T(3) * T(4)$

T	
3	0
4	1
12	2
15	3
1	4
16	5





# Tableaux

---

## Remplir un tableau

Syntaxe de lecture : **Lire** ( Nom\_tab (indice) )

### Exemple:

Remplissage de tous les éléments du tableau T avec l'instruction Lire:

```
Pour i ← 0 à 5 pas 1 Faire
```

```
    Lire ( T (i) ) ;
```

```
Fin Pour
```



# Tableaux

---

## Afficher les éléments du tableau

Syntaxe d'écriture : **Ecrire** ( Nom\_tab (indice) )

### Exemple:

Affichage des valeurs de tous les éléments du tableau T :

```
Pour i ← 0 à 5 pas 1 Faire
```

```
    Lire ( " l'élément ", i+1, "du tableau T est :", T (i) ) ;
```

```
Fin Pour
```



## ***Exercice : Tableaux***

---

Ecrire un algorithme qui permet de demander à l'utilisateur de saisir les notes des étudiants (60 étudiants) , puis l'algorithme calcule et affiche la moyenne des notes.



# *Exercice : Tableaux*

**Algorithme** moyenne\_notes

**Variables**

**Tableau** N (60) : réel

M , S : réel

i : entier

**Début**

$s \leftarrow 0$

**Pour** i  $\leftarrow$  0 à 59 **pas** 1 **faire**

**Ecrire** ("donner la note de l'étudiant num " , i+1, " :");

**Lire** (N (i) );

$s \leftarrow s + N (i)$

**Fin pour**

$M \leftarrow S / 60$

**Ecrire** ("la moyenne des notes est : " , M) ;

**Fin**



## ***Exercice : Tableaux***

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

int main()
{
    float N[3],s,moy;
    int i;
    s=0;
    for(i=0;i<=2;i++){
        printf("donner la note de l'étudiant numéro %d",i+1);
        scanf("%f",&N[i]);
        s=s+N[i];
    }
    moy=s/3;
    printf("la moy est %f",moy);
    return 0;
}
```



## ***Exercice : Tableaux***

---

Ecrire un algorithme qui déclare, remplit et affiche un tableau de 10 entiers en les mettant à zéro.



# Exercice : Tableaux

---

## Algorithme Exercice2

### Variables

**Tableau** T (10) : entier

i : entier

### Début

**Pour** i ← 0 à 9 **pas** 1 **faire**

    T (i) ← 0 ;

**Fin pour**

**Pour** i ← 0 à 9 **pas** 1 **faire**

**Ecrire** (" l'élément ", i+1, " du tableau T est :", T (i) );

**Fin pour**

### Fin



## ***Exercice : Tableaux***

---

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

int main()
{
    float T[10];
    int i;
    for(i=0;i<=9;i++){
        T[i]=0;
    }
    for(i=0;i<=9;i++){
        printf("T[%d]=%d \n",i,T[i]);
    }
    return 0;
}
```





## ***Exercice : Tableaux***

---

Ecrire un algorithme qui déclare, remplit et affiche un tableau contenant les six voyelles de l'alphabet français.



# Exercice : Tableaux

## Algorithme Exercice2

### Variables

**Tableau** T (10) : caractère

i : entier

### Début

T (0) ← " A "

T (1) ← " E "

T (2) ← " I "

T (3) ← " O "

T (4) ← " U "

T (5) ← " Y "

**Ecrire**(" les voyelles de l'alphabet francais sont: " )

**Pour** i ← 0 à 5 **pas** 1 **faire**

**Ecrire** (T (i) );

**Fin pour**

### Fin



## ***Exercice : Tableaux***

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char T[6];
    int i;
    T[0]='A';
    T[1]='E';
    T[2]='I';
    T[3]='O';
    T[4]='U';
    T[5]='Y';
    printf("les voyelles de l'alphabet francais sont: \n");
    for(i=0;i<=5;i++){
        printf("%c \t",T[i]);
    }
    return 0;
}
```



## ***Exercice : Tableaux***

---

Ecrire un algorithme qui demande à l'utilisateur de saisir 10 réels stockés dans un tableau, puis l'algorithme calcule et affiche la somme, le produit et la moyenne des éléments du tableau.



# Exercice : Tableaux

## Algorithme Exercice2

### Variables

**Tableau** T (10) : réel  
som , prod , moy :réel  
i : entier

### Début

som  $\leftarrow$  0

prod  $\leftarrow$  1

**Ecrire**(" entrer les éléments du tableau:" )

**Pour** i  $\leftarrow$  0 à 9 **pas** 1 **faire**

**Lire**( T(i) )

    som  $\leftarrow$  som+T(i)

    prod  $\leftarrow$  prod\*T(i)

**Fin pour**

Moy  $\leftarrow$  som/10

**Ecrire**("la somme des éléments de tableau est :",som);

**Ecrire**("le produit des éléments de tableau est:",prod);


**Ecrire**("la moyenne des éléments de tableau est :",moy);

**FIN**



## ***Exercice : Tableaux***

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
float T[10],som,prod,moy;
int i;
som=0;
prod=1;
printf("entrer les éléments du tableau\n");
for(i=0;i<=9;i++){
    printf("T[%d]=",i);
    scanf("%f",&T[i]);
    som=som+T[i];
    prod=prod*T[i];
}
moy=som/10;
printf("la somme des éléments de tableau est : %f\n",som);
printf("le produit des éléments de tableau est : %f\n",prod);
printf("la moyenne des éléments de tableau est : %f",moy);
return 0;
}
```



## *Exercice : Tableaux*

---

Ecrire un algorithme qui effectue le produit scalaire de deux vecteurs de même taille (3 éléments) représentés par des tableaux à une dimension.

exemple


U(a,b,c)    v(x,y,z)

Le produit scalaire =  $a * x + b * y + c * z$



## ***Exercice : Tableaux***

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
float U[3],V[3];
int i;
float prod;
prod=0;
printf("entrer les valeurs des deux vecteur \n");
for(i=0;i<=2;i++){
    printf("U[%d]= ",i);
    scanf("%f",&U[i]);
    printf("V[%d]= ",i);
    scanf("%f",&V[i]);
    prod=prod + U[i]*V[i];
}
printf("le produit scalaire est : %f",prod);
return 0;
}
```





## ***Exercice : Tableaux***

---

Ecrire un programme qui demande à l'utilisateur de saisir 5 entiers qu'on stock dans un tableau T. ensuite le programme détermine et affiche le minimum et le maximum des éléments du tableau T



```
1  int main()  
2  { int T[5], i, min , max;  
3  printf("donnez les 3 élément du tableau\n");  
4  for(i=0; i<5; i++) {  
5      scanf("%d", &T[i]);  
6  }  
7  min=T[0];  
8  max=T[0];  
9  for(i=1; i<5; i++) {  
10     if(T[i]<min)  
11         min =T[i];  
12     if(T[i]>max)  
13         max =T[i];  
14 }  
15 printf("le max est %d\n", max);  
16 printf("le min est %d\n", min);  
17 return 0;  
18 }  
19
```



## ***Exercice : Tableaux***

```
int main()
{ int T[5],i,min ,max;
printf("donnez les 3 élément du tableau\n");
for(i=0;i<5;i++){
    scanf("%d",&T[i]);
}
min=T[0];
max=T[0];
for(i=1;i<5;i++){
    if(T[i]<min)
        min =T[i];
    if(T[i]>max)
        max =T[i];
}
printf("le max est %d\n",max);
printf("le min est %d\n",min);
return 0;
}
```



---

## Exercice 7

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers dans un tableau T ensuite on redemande à l'utilisateur de saisir un entier N.

L'objectif étant de chercher l'existence du nombre N dans T

Si N se trouve dans T: « N se trouve dans le tableau »

Si N ne se trouve pas dans T: « N ne se trouve pas dans le tableau »



## ***Exercice : Tableaux***

```
int main()
{ int T[5],i,n,ex;
printf("donnez les 3 élément du tableau\n");
for(i=0;i<5;i++){
    scanf("%d",&T[i]);
}
printf("saisir la valeur de n");
scanf("%d",&n);
ex=0;
for(i=0;i<5;i++){
    if(T[i]==n)
        ex++;
}
if (ex==0)
    printf("ne se trouve pas ");
else
    printf("se trouve %d fois ",ex);

return 0;
}
```



```
1  int main()  
2  { int T[5], i, n, ex;  
3  printf("donnez les 3 élément du tableau\n");  
4  for(i=0; i<5; i++) {  
5      scanf("%d", &T[i]);  
6  }  
7  printf("saisir la valeur de n");  
8  scanf("%d", &n);  
9  ex=0;  
10 for(i=0; i<5; i++) {  
11     if(T[i]==n)  
12         ex++;  
13 }  
14 if (ex==0)  
15     printf("ne se trouve pas ");  
16 else  
17     printf("se trouve %d fois ", ex);  
18  
19     return 0;  
20 }  
21
```

---

## Exercice 8

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers dans un tableau T ensuite on redemande à l'utilisateur de saisir un entier N.

Le programme va calculer le nombre d'occurrence de N dans T.



```
1  int main()  
2  { int T[5], i, n, ex;  
3  printf("donnez les 3 élément du tableau\n");  
4  for(i=0; i<5; i++) {  
5      scanf("%d", &T[i]);  
6  }  
7  printf("saisir la valeur de n");  
8  scanf("%d", &n);  
9  ex=0;  
10 for(i=0; i<5; i++) {  
11     if(T[i]==n)  
12         ex++;  
13 }  
14 if (ex==0)  
15     printf("ne se trouve pas ");  
16 else  
17     printf("se trouve %d fois ", ex);  
18  
19     return 0;  
20 }  
21
```



---

## Exercice

Ecrire un programme permettant à l'utilisateur de saisir les notes d'une classe (10étudiants).

Le programme ,une fois la saisie terminée, renvoie le nombre de ces notes supérieures à la moyenne de la classe

