

Exercice 1 : (TD)

Écrivez le sous-programme `afficheNbs` permettant d'afficher sur l'écran des nombres inférieurs à une certaine limite

```
#include <stdio.h>
void afficheNbs(int n){
    int i ;
    for (i=0 ;i<n ;i++)
        printf("%d ",i) ;
}
int main(){
    int x ;
    printf("entrer un nbr\n") ;
    scanf("%d",&x) ;
    afficheNbs(x) ;
}
```

Exercice 2: (TP)

Écrivez le sous-programme `miroir` qui prend un nombre naturel, et affiche son image inversée à l'écran, par exemple 5973, et le programme affichera 3795

```
#include <stdio.h>
void miroir(int n){
    do{
        printf("%d", n % 10) ;
        n=n / 10;
    }while(n>0) ;
}
int main() {
    int x;
    printf("entrer un nbr: ");
    scanf("%d", &x);
    miroir(x);
}
```

Exercice 3 : (TD/TP)

Écrire le sous-programme `afficheTab` pour afficher les éléments d'un tableau

```
#include <stdio.h>
void afficheTab (float t[],int n){
    int i ;
    for (i=0 ;i<n ;i++)
        printf("%f\t",t[i]);
}
int main(){
    float tab[100] ;
    int N,i ;
    printf("entrer le nbr des éléments <=100");
    scanf("%d", &N) ;
    for ( i=0 ;i<N ;i++){
        printf("%d=>",i) ;
        scanf("%f", &tab[i]) ;
    }
    afficheTab (tab,N);
}
```

Exercice 4 : (TD)

Écrire une procédure `max` qui renvoie le max entre 2 nbs reels

- En utilisant une variable globale
- En utilisant le passage par variable
- Réécrire cette procédure sous la forme d'une fonction

```

1 max(float x, float y) {
2   if (x >= y)
3     printf("%f\n", x);
4   else
5     printf("%f\n", y);
6 }

```

```

1 float m;
2 max(float x, float y) {
3   if (x >= y)
4     m = x;
5   else
6     m = y;
7 }

```

```

1 max(float x, float y, float *m) {
2   if (x >= y)
3     *m = x;
4   else
5     *m = y;
6 }

```

```

1 float max(float x, float y) {
2   if (x >= y)
3     return x;
4   return y;
5 }

```

Exercice 5 : (TD/TP)

- Définir une structure pour contenir les coordonnées d'un point (x, y).
- Écrivez un sous-programme « norme » pour calculer la norme d'un vecteur.

```

#include <stdio.h>
#include <math.h>
typedef struct{
    float x,y;
} point;
float norme (point p1, point p2) {
    return sqrt (( p1.x-p2.x) * (p1.x-p2.x) + (p1.y-p2.y) * (p1.y-p2.y) );
}
int main(){
    point p1,p2;
    printf("entrer les coordonnées du 1e point\n");
    scanf("%f%f",&p1.x,&p1.y);
    printf("entrer les coordonnées du 2e point\n");
    scanf("%f%f",&p2.x,&p2.y);
    printf("la norme=%.2f", norme(p1, p2));
}

```

Exercice 6 : (TP)

- Écrivez le sous-programme isSeparateur pour voir si un caractère est un séparateur ou non. Les séparateurs sont (?! , et espace)
- Ecrire un sous-programme countWord pour compter le nombre de mots dans une phrase

```

#include <stdio.h>
#include <string.h>
int isSeparator(char x){
    return x==' '||x==','||
        ||x=='.'||x=='!'||x=='?';
}
int countWord(char s[]) {
    int i, nbWrd, n;
    i = nbWrd = 0;
    n = strlen(s);
    for(i=0;i<n-1;i++)

```

```

    if (!isSeparator(s[i]) && isSeparator(s[i+1]))
        nbWrd++;
    if (n>0 && !isSeparator(s[n-1]))
        nbWrd++;
    return nbWrd;
}
int main(){
    char s[100];
    printf("entrer une phrase\n");
    gets(s);
    printf("le nbr des mots est : %d", countWord(s)) ;
}

```

Exercice 7: (TD)

- Écrivez le sous-programme `fact` pour calculer le factoriel d'un nombre.
- Écrivez un sous-programme `power_1` pour calculer -1 à la puissance de y
- Écrivez un sous-programme `power` pour calculer x à la puissance de y
- Écrivez le sous-programme `cos` pour calculer la somme suivante :

$$S = \sum_{i=0}^n \frac{(-1)^i x^i}{(2i)!}$$

```

#include <stdio.h>
int fact(int y) {
    int I, p = 1;
    for (i = 2; i < y; i++)
        p = p * i;
    return p;
}
int power_1(int y) {
    if (n%2)
        return -1;
    return 1;
}
float power(float x, int y) {
    float p = 1;
    int i;
    for (i = 0; i < y; i++)
        p = p * x;
    return p;
}
float cos(float x, int n) {
    float s = 0;
    int i;
    for (i = 0; i < n; i++)
        s = s * power_1(i) * power(x, i) / fact(i);
    return s;
}
int main(){
    float x;
    printf("entrer un nbr ");
    scanf("%f", &x) ;
    printf("le cosunis=%.2f",cos(x,10000));
}

```

Exercice 8: (TP)

- Si vous savez qu'un nombre parfait est un nombre qui égale a la somme de ses diviseurs sauf 1 et lui-même.
Écrivez le sous-programme `isParfait` pour voir si le nombre est parfait ou non
- Ecrire un programme pour afficher tous les nbrs parfait inferieur à N

```

#include <stdio.h>
int isParfait(int n) {
    int s, i;
    s = 0;

```

```
    for (i = 1; i <= n / 2; i++)
        if (n % i == 0) s += i;
    return n == s;
}
int main(){
    int I,n;
    printf("entrer un nbr ");
    scanf("%d", &n) ;
    printf("voici la liste des nbrs parfaits\n");
    for (i = 1; i <= n ; i++)
        if (isParfait(i))
            printf("%d\t",i);
}
```