



Niveau: 1^e année informatique
Matière: ASD2

Corrige type TD/TP
N° : 02

Année universitaire : 2022/2023
Chapitre 1 : La récursivité

Exercice 1 : (TD)

Soit le programme :

- 0
- La valeur de r ne change pas parce qu'il est passer en valeur
- Il faut le passer en adresse

```
1 #include <string.h>
2 somme(int d, int f, int *s) {
3     int i;
4     *s=0;
5     for (i = d ; i <= f ; i++)
6         *s = *s + i ;
7 }
8 main() {
9     int a,b,r;
10    r=0;
11    printf("entrer deux nombres");
12    scanf("%d%d", &a,&b);
13    somme(a, b, &r);
14    printf("le résultat est %d", r);
15 }
```

```
1 #include <string.h>
2 int somme(int d, int f) {
3     int i, s=0;
4     for (i = d ; i <= f ; i++)
5         s = s + i ;
6     return s;
7 }
8 main() {
9     int a,b,r;
10    r=0;
11    printf("entrer deux nombres");
12    scanf("%d%d", &a,&b);
13    r =somme(a, b);
14    printf("le résultat est %d", r);
15 }
```

Exercice 2 : (TP)

Écrivez un sous-programme récursif *conv10to2* pour convertir du système décimal 10 au système binaire 2 (affichage)

```
1 void conv10to2(int n) {
2     if (n > 0) {
3         conv10to2(n / 2)
4         printf("%d", n % 2);
5     }
6 }
```

Exercice 3 : (TD)

Écrivez un sous-programme récursif *pos* qui renvoie l'emplacement du caractère x dans la chaîne s à partir de la position d et renvoie -1 s'il ne le trouve pas

```
1 int pos(char s[], char x, int d) {
2     if(s[d]=='\0') return -1;
3     if(s[d]== x) return d;
4     return pos(s, x, d+1);
5 }
```

Exercice 4 : (TD/TP)

Écrivez un sous-programme récursif *PGCD* pour calculer le plus grand commun diviseur de deux nombres

Écrire le programme principal pour Réduire la fraction $\frac{A}{B}$ à l'aide de *PGCD*

```
1 int PGCD(int x, int y) {
2     if (y)
3         return PGCD(y, x % y);
4     return x;
5 }
```



```
6 main() {
7   int a, b, d;
8   printf("entrer deux nbrs\n");
9   scanf("%d", &a, &b);
10  d = PGCD(a, b);
11  printf("%d/%d=%d/%d\n", a, b, a/d, b/d);
12 }
```

Exercice 5 : (TP)

Écrivez un sous-programme récursif *racine* pour calculer la racine carrée de a

```
1 float racine(float x, int n) {
2   int i;
3   float r;
4   r = 1;
5   for (i = 0; i < n; i++)
6     r = (r + x / r) * .5;
7   return r;
8 }
1 float racine(float x, int n) {
2   float r;
3   if (n == 0) return 1;
4   r = racine(x, n - 1);
5   return (r + x / r) * .5;
6 }
```

Exercice 6 : (TD/ TP)

Écrivez un sous-programme récursif *Somme* pour calculer la somme suivante :

$$s = \sum_{i=1}^n \frac{-1^{i+1}}{i} x^i$$

```
1 float Somme(float x, int n) {
2   float r;
3   if (n == 0) return 0;
4   if (n%2 == 0) return -1.0/n*pow(x, n)+ Somme(x, n-1);
5   return 1.0/n*pow(x,n)+Somme(x, n-1);
6 }
1 float Somme(float x, int i, int n, float p) {
2   float r;
3   if (i>n) return 0;
4   if (n%2 == 0) return -1.0/i*p+ Somme(x, i+1, n, p*x);
5   return 1.0/i*p + Somme(x, i+1, n, p*x);
6 }
```