REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de M'sila
Faculté des Mathématiques et de l'Informatique
Département d'informatique

جامعة المسيلة
كلية الرياضيات والإعلام الآلي
قسم الإعلام الآلي

**Level:** 1st year computer science
**Material**: ADS2

# Typical solution
# TD/TP N° : 02

**Academic year:** 2023/2024
**Chapter 1** : Recursion

**Exercise 1: (TD)**

Consider the following program:

- 0
- The value of 'r' doesn't change because it uses value passing.
- You need to use address passing."

```c
#include <string.h>
sum(int d, int  f, int *s) {
  int i;
  *s=0;
  for (i =  d; i <= f; i++)
    *s = *s + i;
}
main() {
  int a, b, r;
  r=0;
  printf("enter two numbers");
  scanf("%d%d", &a, &b);
  sum(a,  b, &r);
  printf("the result is %d", r);
}
```

```c
#include <string.h>
  int sum(int d, int f) {
  int i, s=0;
  for (i =  d; i <= f; i++)
    s = s + i;
  return s;
}
main() {
  int a, b, r;
  r=0;
  printf("enter two numbers");
  scanf("%d%d", &a, &b);
  r = sum(a, b);
  printf("the result is %d", r);
}
```

**Exercise 2: (TP)**

Write a recursive subroutine ***conv10to2*** to convert from decimal system 10 to binary system 2 (display)

```c
void conv10to2(int n) {
  if (n > 0) {
    conv10to2(n/ 2)
  printf("%d", n % 2);
  }
}
```

**Exercise 3: (TD/ TP)**

Write a recursive subroutine called 'pos' that returns the location of the character 'x' in string 's' starting from position 'd' and returns -1 if it cannot be found.

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

**Exercise 4: (TP)**

Write a recursive 'PGCD' subroutine to calculate the greatest common divisor of two numbers.

Write the main program to reduce the fraction $\frac{A}{B}$ using the ***PGCD***

```c
int PGCD(int x, int y) {
  if (y)
    return PGCD(y, x % y);
  return x;
}
```

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

**Exercise 5: (TD/TP)**

Write "root", a recursive subroutine to calculate the square root of a

```
1    float root(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 root(float x, int n) {
 2   float r;
 3   if (n == 0) return 1;
 4   r = root(x, n - 1);
 5   return(r + x / r) * .5;
 6 }
```

**Exercise 6: (TD)**

Write "Sum", a recursive subroutine to calculate the following sum: $s = \sum_{i=1}^{n} \frac{-1^{i+1}}{i} x^i$

```
 1 float Sum(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)+  Sum(x, n-1);
 5   return 1.0/n*pow(x,n)+Sum(x, n-1);
 6 }
```

```
1  float Sum(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+ Sum(x, i+1, n, p*x);
 5   return 1.0/i*p + Sum(x, i+1, n, p*x);
 6 }
```