



Level: 1st year computer science
Material: ADS2

Typical solution TD/TP N° : 02

Academic year: 2024/2025
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."

<pre>1 #include <string.h> 2 prod(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("enter two numbers"); 12 scanf("%d%d", &a, &b); 13 prod(a, b, &r); 14 printf("the result is %d", r); 15 }</pre>	<pre>1 #include <string.h> 2 int prod(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("enter two numbers"); 12 scanf("%d%d", &a, &b); 13 r = prod(a, b); 14 printf("the result is %d", r); 15 }</pre>
--	---

Exercise 2: (TP)

Write a recursive subroutine **conv10to8** to convert from decimal system 10 to octal system 8 (display)

```
1 void conv10to8(int n) {
2     if (n > 0) {
3         conv10to8(n/ 8)
4     printf("%d", n % 8);
5     }
6 }
```

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.

```
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 }

int main() {
    char s[100], x;
    int p ;
    printf("Enter a phrase:\n");
    gets(s); // safer than gets, reads string with spaces
    printf("Enter a character:\n");
    scanf(" %c", &x); // Space before %c to consume any stray newline
    p = 0;
    printf("Positions of character '%c' in the string:\n", x);
    p = pos(s, x, p);
    while (p != -1) {
        printf("Found at index: %d\n", p);
    }
}
```



```
p = pos(s, x, p+1);  
}  
return 0;  
}
```

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*

```
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("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 }
```

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