

Higher School of Applied Sciences of Tlemcen
1-year computer science course



Functions

Part 3

Presented by:

Dr. Imane NEDJAR

Part 3

Recursive Functions

Recursion is the technique that allows a function to **call itself**.

Functions

Recursion

Example 1

Write a function that calculates the factorial of a number

```
int factorial (int N){
    int i,f;
    f=1;
    for (i=1;i<=N;i++)
        f=f*i;
    return f;}
```

```
int factorial (int N)
{ if (N!=0)
return N*factorial (N-1);
else return 1;
}
```

Functions

Recursion

Example 1

Write a function that calculates the factorial of a number

```
int factorial (int N)
{ if (N!=0)
return N*factorial (N-1);
else return 1;
}
```

N=4

4!=0

return 4*(factorial (4-1));

3!=0

4*(3* factorial (3-1))

4*(3*(2* factorial (2-1)))

4*(3*(2*(1* factorial (1-1))))

4*(3*(2*(1*1)))=24

Example 1

Write a function that calculates the factorial of a number

```
#include<stdio.h>
int factorial (int N)
{ if (N!=0)
return N* factorial (N-1);
else return 1;}

int main(){
    int N,f;
    printf("Provide the number N\n");
    scanf("%d",&N);
    f= factorial (N);
    printf("The factorial of %d is %d",N,f);
    return 0;}
```

Functions

Recursion

Exemple 2

Write a function that calculates the sum

```
int sum (int N){
    int i,s;
    s=0;
    for (i=1;i<=N;i++)
        s=s+i;
    return s;}
```

$$\sum_{i=0}^N N_i$$

```
int sum (int N)
{ if (N!=0)
return N+sum (N-1);
else return 0;
}
```

Exemple 2

Write a function that calculates the sum of the first 5 numbers

```
int sum (int N)
{ if (N!=0)
return s+sum (N-1);
else return 0;
}
```

N=5

5!=0

return 5+(sum(5-1));

4!=0

5+(4+sum (4-1))

5+ (4+ (3+sum (3-1)))

5+ (4+ (3+ (2+sum (2-1))))

5+ (4+ (3+ (2+ (1+sum (1-1)))))

5+ (4+ (3+ (2+ (1+0))))

Exercise 1

The sequence defined as follows:

$$\begin{cases} u_{n+1} = u_n + 5 \\ u_0 = 1 \end{cases}$$

-Create a recursive function to compute the nth term of this sequence.

-Invoke this function with n=4.