

EXPERIMENT 3

Recursive Programming Using C

Experiments:

REMINDER:

- Every recursive function must have one or more base cases,
 - The general (recursive) case must eventually be reduced to a *base case* (terminating condition),
 - The base case stops the recursion,
 - Every recursive call has its own copy of parameters and the local variables ,
 - A *recursive function* is a function that calls itself .

Experiment 1:

Let `a` be an array of integers. Present recursive algorithms and write their C implementations to compute:

- the maximum (minimum) element of the array,
- the sum (product) of the elements of the array,
- the average of the elements of the array

HINT: Use functions having the following prototypes:

```
<return type> name_of_the_function(int a[ ], int n);
```

Experiment 2:

Consider the following recursive algorithm (pseudocode)

```
int fun1(x)
{
    if(x < 5)
        return (3 * x)
    else
        return (2 * fun1(x - 5) + 7)
}
```

What would be returned if `fun1` is called as `fun1(12)`? Perform analysis of the problem manually, **without using computer**.