FONDAMENTI DI INFORMATICA

RECURRENCE

Prof. Luigi Maria Ingrosso
Luigi.Maria.Ingrosso@uniroma2.it
Concept: Basic Stack Operations

- Push
- Pop
- Peek
- Check
Concept: Basic Stack Operations

- The **Activation Stack** is a stack used by the operating system to manage the execution of function calls.
- Each function stores in the stack:
  - the location in memory of the function to be evaluated
  - the return address (location in memory to return when the function terminated)
  - the copy of function parameters
  - the space to store local variable
Function call

Application A activation stack

A call f1
A is suspended

f1 call f1.1
f1 is suspended
f1.1 terminate and A call f2
f1 restart and is immediately suspended

f2 and f1 terminate
A restart

A call f5
A is suspended
Concept: The Activation Stack

- When the user starts a program, the operating system allocates enough memory to load the application and a block of memory to contain its activation stack.

- When that application calls a function, the parameters and local variables occupy a stack frame that is pushed onto the activation stack.

- The calling program is then suspended and control is passed to the function specified in the new frame.

- When that function completes, its frame is destroyed; any returned data and control is returned to the frame beneath.

- Note that in a MATLAB implementation, the stack frame is the storage environment for the current workspace.
Function Specification vs Function Instances

• The file defining the function merely specifies how it would behave if you ever called it. This is the function specification.

• When you call the function, a stack frame is created, parameter values are supplied and you have a fully defined instance of that function.

• There is no reason in principle why a function cannot “call itself,” because each call is a different instance of the function.
Example of a function that call itself

- printwords is a function that print a sentence from the latest word to the first, e.g.

```python
>>> printwords('what does this do')
do
does
this
what
```
function printworlds(sent)
    [word, rest] = strtok(sent);
    if ~isempty(rest)
        printwords(rest)
    end
    disp(word)
end
9.2 Recursion Defined

For a recursive program to succeed, it must satisfy three basic requirements:

• There must be a terminating condition specifying when to stop the recursion

• The function must call a clone of itself (with different parameters)

• The parameter change must move the data towards the terminating condition
Implementing a Recursive Function in MATLAB

The MATLAB template for recursion is:

```matlab
function <result> = <function_name> (<formal_params>)
% <documentation>
if <terminating condition 1>
    <result> = <initial value 1>
elseif <terminating condition 2>
    <result> = <initial value 2>
. . .
else
    <result> = <operation> ...
    (<formal_params>, ... , <formal_params>, ... , \n    <function_name> (<new_params>) )
end
```
Factorial

- \( n! = n \times (n-1)! \), \( 0! = 1 \) (by def)
  - \( 3! = 3 \times 2! = 3 \times 2 \times 1! = 3 \times 2 \times 1 \times 0! = 3 \times 2 \times 1 \times 1 = 6 \)

function result = fact(N)
% recursive computation of N!
% fprintf('fact( %d )\n', N); % testing only
    if N == 0
        result = 1;
    else
        result = N * fact(N - 1);
    end
**fact(3)**

- \(N=3, \text{fact}(3)\),  \(\text{result} = 6\)
- \(N=2, \text{fact}(2)\),  \(\text{result} = 2\)
- \(N=1, \text{fact}(1)\),  \(\text{result} = 1\)
- \(N=0, \text{fact}(0)\),  \(\text{result} = 1\)
The Fibonacci Series

- Leonardo Pisano Fibonacci, studied rabbit populations
- Starting with a pair of newborn rabbits, he wanted to calculate the rabbit population after a year.
  \[ \text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2) \]
- This is, of course, recursive where \( \text{fib}(1) \) and \( \text{fib}(2) \) are both 1.
- Computationally, this turns out to be very unpleasant
- Results in the series:
  - 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...
- This series occurs frequently in nature
Fibonacci Rabbits
Fibonacci in Nature
fib(5)

- N=5, fib(5)=fib(4)+fib(3),
- N=4, fib(4)=fib(3)+fib(2),
- N=3, fib(3)=fib(2)+fib(1),
  - N=2, fib(2)=1
  - N=1, fib(1)=1
  - N=3, result=2
- N=2, fib(2)=1
- N=4 result=3
- N=3....
- N=5 result=5
Palindromes

- alla
- onorarono
- Ingegni
- radar
- anna
-otto
- parlo col rap
- i topi non avevano nipoti
function ans = isPal(str)
% recursive palindrome detector
    if length(str) < 2
        ans = true;
    elseif str(1) ~= str(end)
        ans = false;
    else
        ans = isPal(str(2:end-1));
    end