



ΕΡΓΑΣΤΗΡΙΟ 8

Σκοπός του 8^{ου} εργαστηρίου είναι η εκμάθηση και εξοικείωση με τις αναδρομικές συναρτήσεις αλλά και η χρήση στοίβας. Στη συνέχεια ακολουθούν δυο παραδείγματα αναδρομικών συναρτήσεων: factorial (παραγοντικό) και Fibonacci.

C	Assembly
<pre>int factorial(int n){ int f; if(n <= 0){ f = 1; }else{ f = n * factorial(n-1); } return f; }</pre>	<pre>factorial: # \$a0 = n # \$v0 = f addi \$sp, \$sp, -8 # space for two words sw \$ra, 4(\$sp) # save return address sw \$a0, 0(\$sp) # temporary variable to hold n li \$v0, 1 # f = 1 ble \$a0, \$zero, factorial_return # if n <= 0 addi \$a0, \$a0, -1 # n = n - 1 jal factorial lw \$a0, 0(\$sp) # retrieve original n mul \$v0, \$v0, \$a0 # f = n * factorial(n-1) factorial_return: lw \$ra 4(\$sp) # restore \$ra addi \$sp, \$sp, 8 # restore \$sp jr \$ra # back to caller</pre>



C	Assembly
<pre>int fibonacci(int n){ int f; if(n < 2){ f = n; }else{ f = fibonacci(n-1) + fibonacci(n-2); } return f; }</pre>	<pre>fibonacci: # \$a0 = n # \$v0 = f addi \$sp, \$sp, -12 # space for three words sw \$ra, 0(\$sp) sw \$s0, 4(\$sp) sw \$s1, 8(\$sp) add \$s0, \$a0, \$zero addi \$t1, \$zero, 1 beq \$s0, \$zero, return0 # if (n == 0) beq \$s0, \$t1, return1 # if (n == 1) addi \$a0, \$s0, -1 jal fibonacci # v0 = fibonacci(n-1) add \$s1, \$zero, \$v0 # s1 = fibonacci(n-1) addi \$a0, \$s0, -2 jal fibonacci # v0 = fibonacci(n-2) add \$v0, \$v0, \$s1 # v0 = fibonacci(n-2) + \$s1 exitfib: # read registers from stack lw \$ra, 0(\$sp) lw \$s0, 4(\$sp) lw \$s1, 8(\$sp) addi \$sp, \$sp, 12 # bring back stack pointer jr \$ra return0: li \$v0, 0 j exitfib return1: li \$v0, 1 j exitfib</pre>



ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018 – 2019

```
printf("The sequence of moves involved in the Tower of Hanoi are :\n");  
towers(num, 1, 3, 2);  
  
return 0;  
}  
  
void towers(int num, int frompeg, int topeg, int auxpeg){  
    if(num == 1){  
        printf("\nMove disk 1 from peg %d to peg %d", frompeg, topeg);  
  
        return;  
    }  
  
    towers(num - 1, frompeg, auxpeg, topeg);  
  
    printf("\nMove disk %d from peg %d to peg %d", num, frompeg, topeg);  
  
    towers(num - 1, auxpeg, topeg, frompeg);  
}
```

Το πρόγραμμά σας κατά την εκτέλεση θα πρέπει να έχει την ακόλουθη μορφή:

Enter the number of disks : 3

The sequence of moves involved in the Tower of Hanoi are :

Move disk 1 from peg 1 to peg 3

Move disk 2 from peg 1 to peg 2

Move disk 1 from peg 3 to peg 2

Move disk 3 from peg 1 to peg 3

Move disk 1 from peg 2 to peg 1

Move disk 2 from peg 2 to peg 3

Move disk 1 from peg 1 to peg 3



Υπενθύμιση

Τα προγράμματα που θα παραδίδετε, θα βαθμολογούνται για την ορθότητα τους αλλά και για την σωστή μορφοποίηση (στοίχιση, σχόλια, κλπ.).

Σχετικά με την παράδοση των ασκήσεων

Η παράδοση της εργαστηριακής άσκησης θα πρέπει να γίνει μέχρι την **Τρίτη 16/4** και η εξέτασή της θα πραγματοποιηθεί την **Τετάρτη 17/4** και ώρα αναλόγως του εργαστηρίου στο οποίο είστε εγγεγραμμένοι.

Σε περίπτωση απουσίας του φοιτητή από το εργαστήριο είναι υποχρεωτικό να σταλούν οι ασκήσεις σε ηλεκτρονική μορφή. Για την υποβολή των ασκήσεων, θα δημιουργήσετε ένα φάκελο με όνομα, το όνομα, το επώνυμο και τον ΑΜ σας, πχ. ονομα-ερωνηγο-211xxxx, και θα αντιγράψετε σε αυτόν μόνο τα δύο .s αρχεία. Στην συνέχεια θα συμπιέσετε τον φάκελο (rar ή zip). Τέλος θα ανεβάσετε τον συμπιεσμένο φάκελο στο eclass.