



Computer Science 3B
Practical Assignment E2
2017-11-02

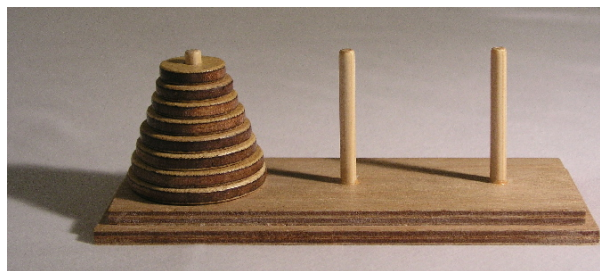
Deadline - 2017-11-03 16h00

Marks: 40

This practical assignment must be uploaded to eve.uj.ac.za **before** 2017-11-03 16h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present. See the reminder page for more details.

Write an x86 assembly program that illustrates the execution of the classic *Towers of Hanoi* game. The Towers of Hanoi requires a player to move a stack of n discs (in decreasing size) from one peg to another peg, where a larger disc cannot be placed on top of a smaller disc.



In the case of the included picture the discs can be labeled 1,2,3,...,n where disc 1 is the smallest and disc n is the largest. The pegs can be labeled A, B, C from left to right. All the discs must be moved from Peg A to Peg C, where one disc can be moved at a time, and a larger disc cannot be put on top of a smaller disc.

This problem can be solved in a recursive manner (generalised for n discs, base case not shown):

1. Move $n - 1$ discs from peg A to peg B
2. Move disc n from peg A to C
3. Move $n - 1$ discs from peg B to peg C

The following steps are repeated for each disc to be moved. A recursive function which realises the above procedure will have the following prototype (Written in C):

```
int hanoi(int disc, char* from, char* to, char* spare);
```

The output of your application should be as follows (12 discs):

```
Enter the number of discs: 12
Move disc 1 from peg A to peg B
Move disc 2 from peg A to peg C
Move disc 1 from peg B to peg C
Move disc 3 from peg A to peg B
Move disc 1 from peg C to peg A
Move disc 2 from peg C to peg B
Move disc 1 from peg A to peg B
Move disc 4 from peg A to peg C
Move disc 1 from peg B to peg C
Move disc 2 from peg B to peg A
Move disc 1 from peg C to peg A
Move disc 3 from peg B to peg C
...
Move disc 3 from peg B to peg C
Move disc 1 from peg A to peg B
Move disc 2 from peg A to peg C
Move disc 1 from peg B to peg C
There were 4095 function calls
```

The output of your application should be as follows (3 discs):

```
Enter the number of discs: 3
Move disc 1 from peg A to peg C
Move disc 2 from peg A to peg B
Move disc 1 from peg C to peg B
Move disc 3 from peg A to peg C
Move disc 1 from peg B to peg A
Move disc 2 from peg B to peg C
Move disc 1 from peg A to peg C
There were 7 function calls
```

No design is required!

Mark sheet

1. Move discs between correct pegs in first step [10]
 2. Move discs between correct pegs in second step [10]
 3. Correct execution. [20]
-

NB

Submissions which **do not assemble** will be capped at 40%!

Execution marks are awarded for a correctly functioning application and not for having some related code.

Reminder

Your submission must follow the naming convention as set out in the general learning guide.

Example

Surname	Berners-Lee	Initials	TJ
Student number	209912345	Module Code	CSC3B10
Current Year	2017	Practical number	PE2

Berners-Lee_TJ_209912345_ CSC3B10_2017_PE2

Your submission must include the following in a single zip (compressed) file:

- *Source file* (asm file) - File containing your solution. Your details must be included at the top of the source code.
- *Program design* (pdf file) - File containing your design. Your details must be included at the top of the design.