

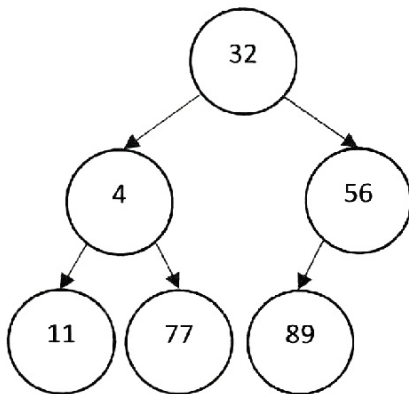
Main V 2

1. Download MARS tool for
https://courses.missouristate.edu/KenVollmar/mars/MARS_4_5_Aug2014/Mars4_5.jar
2. Complete MARS tutorial Part 1 <https://courses.missouristate.edu/KenVollmar/mars/CCSC-CP%20material/MARS%20Tutorial.doc>
3. Implement "Hello world" with MIPS assembly. The program prints "Hello, world!" on the screen.
4. Implement an assembly program which uses a recursive procedure to go through a binary tree in a depth-first order.
The program takes an array as input, where nodes of a binary tree are stored by levels, for example:

.data

sequence: .word 32, 4, 56, 11, 77, 89

is equivalent to a tree:



The program prints the nodes of the tree in depth-first order (starting from the left) as output, for example:

11 77 4 89 56 32 – for the tree above

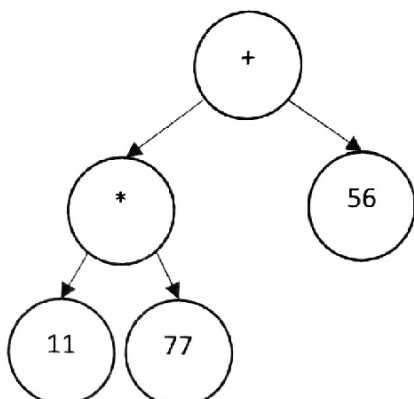
A number of nodes in the tree (a length of the array) is not limited, but known and can be used as a constant in the code.

5. Extend the program from 4 to calculate an arithmetic expression stored as a tree in an array.
Only +, - and * are supported, for example:

.data

sequence: .word '+', '*', 56, 11, 77

is equivalent to a tree:



which represents an arithmetic expression:

$$(11 * 77) + 56$$

Numbers 42, 43 and 45 cannot be used in the expression as they encode the arithmetic operations in ASCII.

The program prints a single number - a result of calculation – as output.

6. Provide 2 hard copies of source .asm files with implemented programs - for tasks 4 and 5. Also, provide a test example for each of 2 programs – with an input array with a tree and output of your program.