EXPRESSION COMPILER USING JAVA AND ASSEMBLY

In main it takes a file named ***example.co*** to read and creates an output file named ***example.asm*** to write. And then it calls a function called ***program()*** and defines input and output files.

Program() read input file into 'str' string, writes beginning and ending of the out file and calls stmt(str).

FUNCTIONS

stmt(String exp) it takes whole statement and processes it as an exp than sends exp to expr.

expr(String exp) takes exp as a expr finds first + that is not in between '(' and ')' sends first part to term second part to moreterm if there isn't a + that is not in between '(' and ')' than sends exp to term

moreterm(String exp) removes + sends remaining to expr and writes code for add.

term(String exp) takes exp as a term finds first * that is not in between '(' and ')' sends first part to factor second part to morefactor if there isn't a * that is not in between '(' and ')' than sends exp to factor.

morefactor(String exp) removes * sends remaining to term and writes code for multiplication.

factor(String exp) if exp is like (*) it removes '(' and ')' sends remaining to expr. Else if exp is like pow(*) it removes 'pow(' and ')' sends remaining to power operation mypower() and writes code for push-num exp. Else if exp is a hexadecimal number it writes code for push-num exp. Else if exp is an int it writes code for push-num exp. Else it writes code for push-val-var.

There is also an ***isInt(String s)*** function which returns true if s only consists of digits.There is also an ***mypower(int a, int n)*** function which returns the result of power operation.