Factoring Toolbox

# Purpose of the software:

The software package *facroring.py* allows the user to enter any expression and have it returned back completely factored or a statement telling the user it can’t be factored.

Example: x2 + 3x + 2 factors to (x+2)(x+1)

# Instructions for using the software:

1. Create and save a new Python file in the same directory as *factoring.py*
2. Import the function *factor* from the package *factoring* by typing:  
   from factoring import factor
3. Call the function *factor* with the argument of the expression to be factored. The spacing in the expression does not matter. All exponents should be done using carets (eg. x^2). Any variable may be used, not just x.

# Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **What it returns** | **Programmer** |
| factor() | expr | The fully factored expression | Jacob |
| commonFactorCoeffs() | coeffs | The common factored coefficients along with the coefficient taken out | Justin |
| factorNegative() | coeffs | The coefficients with a negative factored out | Jacob |
| trinomialFactor() | coeffs, varis | The two factors of the trinomial or “dnf” if it doesn’t factor | Simon |
| diffOfSquares() | coeffs, varis | The two factors of the Difference of squares expression. Uses trinomialFactor(). | Justin |
| getTerms() | expr | An array of all terms in the expression. | Justin |
| getCoeffAndVar() | term | A tuple of the coefficient and variable of the given term | Jacob |
| getMultiples() | num | An array of tuples of all possible multiples of the given number | Jacob |
| findVarI() | term | The first index of a variable in the given term | Jacob |
| getPositives() | coeffs | A copy of the given array of coefficients where all numbers are positive | Simon |
| getStr() | num | The string version of a number with its sign | Simon |
| formatIn() | Input | The given expression with all spaces taken out of it | Jacob |
| formatOut() | output | The given out output with all “x^1”’s changed to “x” and all “1x”’s changed to “x” | Jacob |
| isValid() | expr | An empty string if the expression if valid an error message for the user if it is not | Jacob |
| commonFactorVaris() | varis | The common factored variables | Jacob |

# Test Cases:

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected output** | **Working?** |
| **Normal** | | |
| 4x^2-9 | (2x+3)(2x-3) | Yes |
| 4x^2 + 12x + 9 | (2x+3)^2 | Yes |
| 4y^2-12y+9 | (2y-3)^2 | Yes |
| -4x^2+12x-9 | -(2x-3)^2 | Yes |
| 8x^2-14x-15 | (4x+3)(2x-5) | Yes |
| x^2+x-12 | (x-3)(x+4) | Yes |
| 4x^2+12x-9 | Does not factor | Yes |
| 7x^2-2x+16 | Does not factor | Yes |
| 2x^2+4x+6 | 2(x^2+2x+3) | Yes |
| 8x^2-8x-6 | 2(2x+1)(2x-3) | Yes |
| 10x^2-15x-25 | 5(2x-5)(x+1) | Yes |
| **Extra** | | |
| 81x+27y-9z | 9(9x+3y-z) | Yes |
| 3x^3-5x^5+6x^2 | x^2(3x-5x^3+6) | Yes |
| **Invalid** | | |
| y=3x-2+7y | Must be expression | Yes |
| 3x | Must have more than one term | Yes |
| 5, + 6t; | Invalid characters | Yes |