Hayes Crowley

Prof. Papademas

CSC\_204\_LAB\_09

**SOURCE CODE**#Evaluate the expression: 4x^2 - 8x + 2

.data

prompt: .asciiz "Enter an input for x to evaluate 4x^2 - 8x + 2: "

output: .asciiz "The output is "

isPositiveString: .asciiz "The result is positive"

isNegativeString: .asciiz "The result is negative"

.text

main:

li $v0, 4 #output the prompt for user

la $a0, prompt

syscall

li $v0, 5 #input the number and save it to $s0

syscall

move $s0, $v0

#square the input

#lw $t0, ($s0)

mul $t0, $s0, $s0

#multiply by 4

mul $t0, $t0, 4

#multiply input by -8

#lw $t2, ($s0)

mul $t2, $s0, -8

#add it all

add $t0, $t0, $t2

add $t0, $t0, 2

move $s2, $t0

li $v0, 4 #print the output label

la $a0, output

syscall

li $v0, 1 #output the number that was entered

move $a0, $s2 #could also use lw $a0, $s0. Pseudo code

syscall

jal printNewLine

# branch-and-link if less than zero

bltzal $s2, isNegative

# branch-and-link if greater than zero

bgezal $s2, isPositive

b exit

printNewLine:

li $v0, 0xB

#print newline, 0xB = 11

#which is the system call to print a single character

la $a0, 0xA #0xA is the newline char

syscall

jr $ra

isPositive:

li $v0, 4

la $a0, isPositiveString

syscall

jr $ra

isNegative:

li $v0, 4

la $a0, isNegativeString

syscall

jr $ra

exit:

li $v0, 10

syscall

**OUTPUT**

**Text

Description automatically generated**

**LAB QUESTIONS**

1. **Explain what is accomplished by this assembly language statement, taken from the original starter code for this application.**

**move $s0, $v0**  
This statement simply moves the value in the function return register $v0 into the saved temporary register $s0 to be used later when calculating the polynomial function.

1. **4 *x* 2 − 8 *x* + 2**  is classified as a polynomial because it is in standard polynomial form, in which the term with the highest degree is placed first, followed by the term with the next lowest degree, and so forth. This expression is a polynomial with a degree of 2.
2. **Screenshot of plot of y = 4 *x* 2 − 8 *x* + 2**  **from Wolfram Alpha:**

**Graphical user interface

Description automatically generated with medium confidence**

1. Within the starter code, the newline character 0xA (LF or line feed) is the hex code for looking up the new line character in ASCII.
2. There was only one prompt for user input in the original starter code.