**PROJECT Introduction to the MIPS Assembler ( Part 3 )**

**Objective** To examine various topics concerning the assembly language.

***PROJECT DESCRIPTION***

Construct and / or modify a program that allows for user input within a MIPS program.

Your program is to evaluate a given polynomial expression when a value is entered for the underlying variable.

***Information About this Project***

Complete / modify the asm program that allows for the evaluation of a quadratic polynomial.

For the starter code, this polynomial is given:

4 *x* 2 − 8 *x* + 2

If, for example, the user of our program enters 6 for *x* , then the program output should be:

4 ( 6 ) 2 − 8 ( 6 ) + 2

or

98

***Steps to Complete this Project***

**STEP 1 Run a Program in MARS**

Open MARS and go to your menu and start a new assembly ( .asm ) file by clicking on [ File ] > [ New ] .

Paste in the code that follows.

**STEP 2 Run and Test Your program**

Test the asm program with user input.

**STEP 3 Modify Your program**

Modify the program that will test whether or not the expression evaluates to a negative or positive number.

**STEP 4 Show Resulting Runs**

Snapshot 2 sample run results ( one for showing a positive evaluation and the other showing a negative result ) into MS Word along with your modified source for credit.

Then, answer each of the questions that appear at the end of this project.

**PROJECT Introduction to the MIPS Assembler ( Part 3 )**

**Figure 1 Program for the Evaluation of a Polynomial**

|  |
| --- |
| #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 "  .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    li $v0, 4 #print the output label  la $a0, output  syscall    li $v0, 1 #output the number that was entered  move $a0, $t0 #could also use lw $a0, $s0. Pseudo code  syscall    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    li $v0, 10 #exit program  syscall |

**PROJECT Introduction to the MIPS Assembler ( Part 3 )**

**STEP 5 Questions and Answers Concerning this Computer Laboratory Project**

Open MS Word and, within a new document, place your responses to these questions. Submit your completed MS Word document for credit.

**(1) ( Assembly Language: Polynomial Evaluation )**

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

move $s0, $v0

**(2) ( Assembly Language: Polynomial Evaluation )**

For the starter code, this polynomial is given.

4 *x* 2 − 8 *x* + 2

Why is this expression classified as a polynomial?

**(3) ( Assembly Language: Polynomial Evaluation )**

The modification, discussed earlier, for this application, requires a test of whether the polynomial evaluates as a positive or negative number.

Visit this Web site:

[**https://www.wolframalpha.com**](https://www.wolframalpha.com)

Enter the following command at their computational engine.

plot y = 4x^2 - 8x + 2

Observe the resulting plot, which shows the range of values for which the polynomial evaluates to a negative number.

Use a Snipping Tool to take a screen snapshot of the plot.

**(4) ( Assembly Language: Polynomial Evaluation )**

Within the starter code, what is the newline character that was used?

**(5) ( Assembly Language: Polynomial Evaluation )**

How many assembly language prompt statements were used in the original starter code for this application?