!Variables and Macros

define(x\_r, l1)

define(y\_r, l2)

define(y\_max, l3)

!Change the following to test further

define(a1, 3)

define(a2, 10)

define(a3, 15)

.section ".data"

outputString: .asciz "For the input of x = %d, y = %d \n"

ymaxString: .asciz "The maximum output value is y = %d, when input x = %d."

.align 4

.section ".text"

.global main

main:

save %sp, -96, %sp

mov -4 , %x\_r

mov 0, %y\_max

ba loop

nop

!controls the y\_max value

compare:

mov %y\_r, %y\_max

ba here

nop

loop:

!3x^4-10x^2-15x

!x^4

mov %x\_r, %o0

call .mul !x^2

mov %x\_r, %o1

call .mul !x^3

mov %x\_r, %o1

call .mul !x^4

mov %x\_r, %o1

call .mul

mov a1, %o1 !3x^4

mov %o0, %y\_r

mov %x\_r, %o0

call .mul !x^2

mov %x\_r, %o1

call .mul

mov a2, %o1 !10x^2

sub %y\_r, %o0, %y\_r !3x^4 - 10x^2

mov %x\_r, %o0

call .mul !15x

mov a3, %o1

sub %y\_r, %o0, %y\_r !3x^4 - 10x^2 - 15x

!Printing

mov %x\_r, %o1

set outputString, %o0

call printf

mov %y\_r, %o2

mov %x\_r, %o2 !move x\_r to o2 for printing later.

!Save largest value

cmp %y\_r, %y\_max

bg compare

nop

here:

inc %x\_r !increment x\_r

cmp %x\_r, 6

ble loop

nop

set ymaxString, %o0

call printf

mov %y\_max, %o1

ret

restore