.data #define segmento de memoria de dados

b: .word 0 #valor inteiro a escolha para testar o codigo

d: .word 0 #valor inteiro a escolha para testar o codigo

e: .word 0 #valor inteiro a escolha para testar o codigo

c: .word 0 #zera valor de c

.text #define inicio segmento de texto

.globl main #define inicio do codigo

main: #label (endereco simbolico) que define inicio do codigo

# Leitura de b

li $v0, 5 # Código de serviço para leitura de inteiro

syscall # Chama o sistema para ler o inteiro

add $s0, $v0, $zero # Move o valor lido para $s0 (b)

sw $s0, b # Armazena o valor de b na memória

# Leitura de d

li $v0, 5 # Código de serviço para leitura de inteiro

syscall # Chama o sistema para ler o inteiro

add $s1, $v0, $zero # Move o valor lido para $s1 (d)

sw $s1, d # Armazena o valor de d na memória

# Leitura de e

li $v0, 5 # Código de serviço para leitura de inteiro

syscall # Chama o sistema para ler o inteiro

add $s2, $v0, $zero # Move o valor lido para $s2 (e)

sw $s2, e # Armazena o valor de e na memória

addi $t0, $s0, 0x23 # a = b + 35, onde a = $t0, b = $s0 e 35 escrito em hexa

add $t1, $t0, $s2 # $t1 recebe a soma de a com e (a + e)

# iniciando valores para calcular d^2 e d^3.

addi $t2, $s1, 0 # $t2 recebe d para usar no decrementador

addi $t3, $zero, 0 # Zera um temporário ($t3) para usar nas somas (d^2)

loop\_square:

add $t3, $t3, $s1 # $t3 recebe d + d (soma de d)

addi $t2, $t2, -1 # Decrementa $t2

bne $t2, $zero, loop\_square # Compara e se acumulador não chegou no valor de d, desvia pro loop de soma

# Início do segundo loop para calcular d^3 com base em d^2

addi $t2, $s1, 0 # Reinicia $t2 com d para o decrementador

addi $t4, $zero, 0 # Inicia $t4 para realizar as somas (d^3)

loop\_cube:

add $t4, $t4, $t3 # $t4 = $t4 + d^2 (acumula d^2)

addi $t2, $t2, -1 # Decrementa $t2

bne $t2, $zero, loop\_cube # Enquanto $t2 != 0, continua no loop

sub $s3, $t4, $t1 # c = d^3 - (a + e)

sw $s3, c # Armazena o resultado final (c) na memória.