**TAREFA FEITA POR**

**Allan Kevin Scain**

[Felipe Veiga dos Santos](mailto:felipesantos@sou.faccat.br)

**Gabriel Krummenauer de Morais**

**1. Considere o seguinte fragmento de código expresso na linguagem C:**

**...**

**r = 0;**

**i = 0;**

**while (i < 10) {**

**if (X[i] > 0) {**

**r = r + X[i];**

**}**

**i++;**

**}**

**…**

Utilize a pseudo-instrução .**word** para alocar memória para as variáveis **r** e **i**, bem como para o vetor **X** com 10 elementos.

**a) Escreva um código em linguagem de montagem do MIPS que realize este trecho de programa em C, explicando cada instrução empregada.**

...

.data

varR: .word 0 # alocando R

varI: .word 0 # alocando I

varX: .word 1 1 1 1 -1 -1 1 1 1 1 # alocando vetor X

.text

li $s0, 0 # salva o valor 0 para uso futuro

while: # laço principal do programa

lw $t0, varI($zero) # armazena o valor de varI em $t0

bge $t0, 10, saida # enquanto $t0(I) >= 10 continua no laço

mul $t1, $t0, 4 # faz a multiplicação para leitura do array de 4 em 4

lw $t2, varX($t1) # armazena em $t2 o valor correspondente ao índice

ble $t2, $s0, condIf # direciona para condicional if se X[I] <= 0

lw $t3, varR($zero) # armazena R em $t3

add $t3, $t3, $t2 # R = R + X[I]

sw $t3, varR($zero) # atualiza R na memoria

condIf:

addi $t0, $t0, 1 # incrementa o I

sw $t0, varI($zero) # atualiza I na memória

j while # retorna para o início do laço

saida: # imprime o valor de R

li $v0, 1 # configura saída para inteiro

lw $t3, varR($zero) # armazena R em $t3

move $a0, $t3 # move valor de $t(R) a ser escrito

syscall

…

Resultado: 8

Tambem disponivel em .asm

**b) Quantas instruções no máximo são executadas em uma dada execução do programa?**

exatamente 115 instruções, sendo elas 110 dentro do laço while, uma antes e 4 depois

O laço em si, com uma constante, pode ser reduzido para 100 execuções, sem instruções de saída do resultado.

obs: Optamos por utilizar o 0 em um registrador ($s), por ser um valor que não muda pode ser usado mais eficientemente na execução do if.