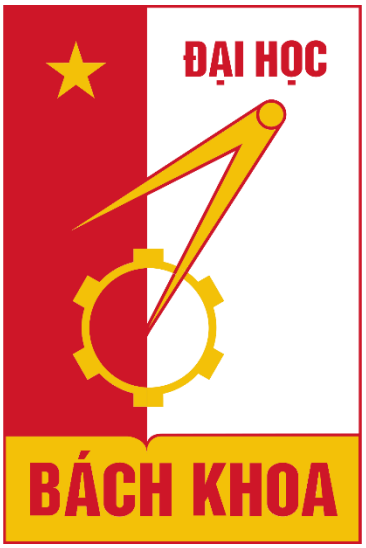
ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO MÔN HỌC

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

ĐỀ BÀI

MID TERM

Giảng viên hướng dẫn: ThS. Lê Bá Vui

Sinh viên: Võ Tá Hoan

MSSV: 20194568

Hà Nội, 20 tháng 05 năm 2022

Nội dung

Assignment 1

3 cạnh của tam giác luôn phải thỏa mãn điều kiện là tổng của 2 cạnh luôn >= cạnh còn lại

Ta sử dụng lệnh blt để kiểm tra

Thuật toán :

if(a+b<c) return 0;

if(b+c<a) return 0;

if(c+a <b) return 0;

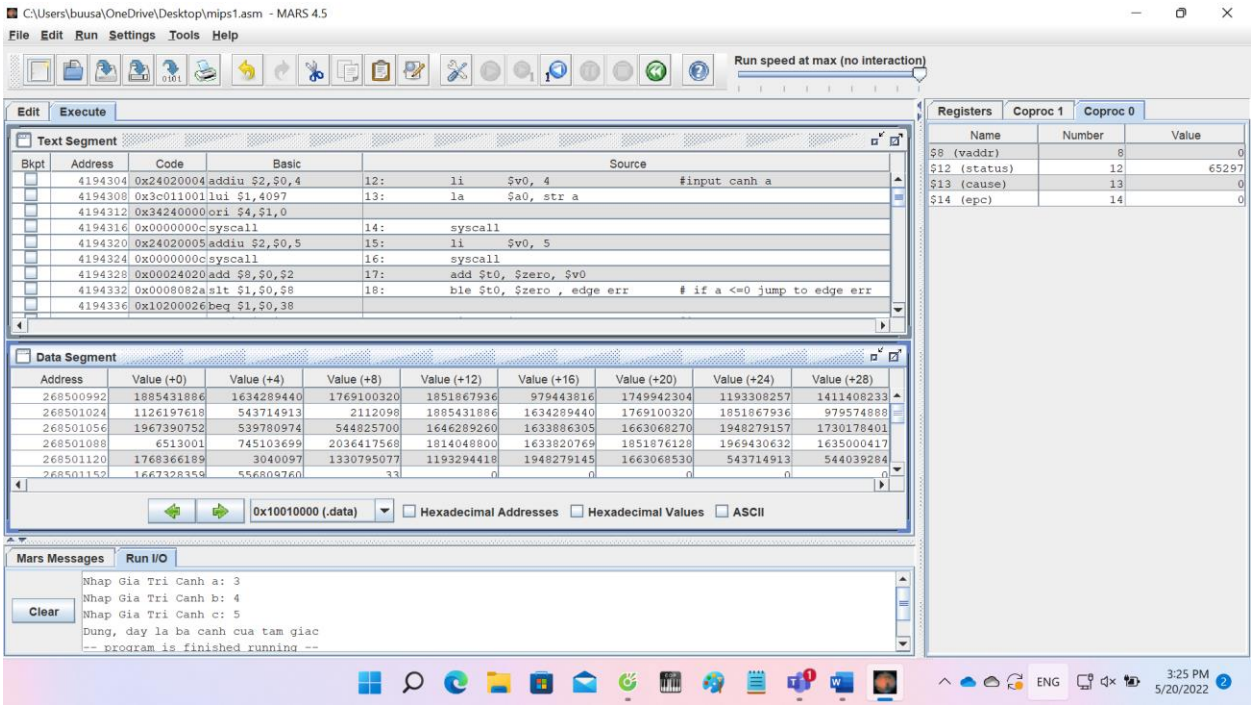
return 1;

```
        .data
str_a: .asciiz "Nhap Gia Tri Canh a: "
str_b: .asciiz "Nhap Gia Tri Canh b: "
str_c:  .asciiz "Nhap Gia Tri Canh c: "
yes:        .asciiz "Dung, day la ba canh cua tam giac"
no:         .asciiz "Sai, Day 0 la ba canh cua tam giac."
err:   .asciiz "ERROR: Gia tri canh tam giac >0!!"

         .text
main:   # Main program code here

        li     $v0, 4                       #input canh a
        la     $a0, str_a
        syscall
        li      $v0, 5
        syscall
        add $t0, $zero, $v0
        ble $t0, $zero , edge_err  # if a <=0 jump to edge_err

        li     $v0, 4               #input canh b
        la     $a0, str_b
        syscall
        li      $v0, 5
```

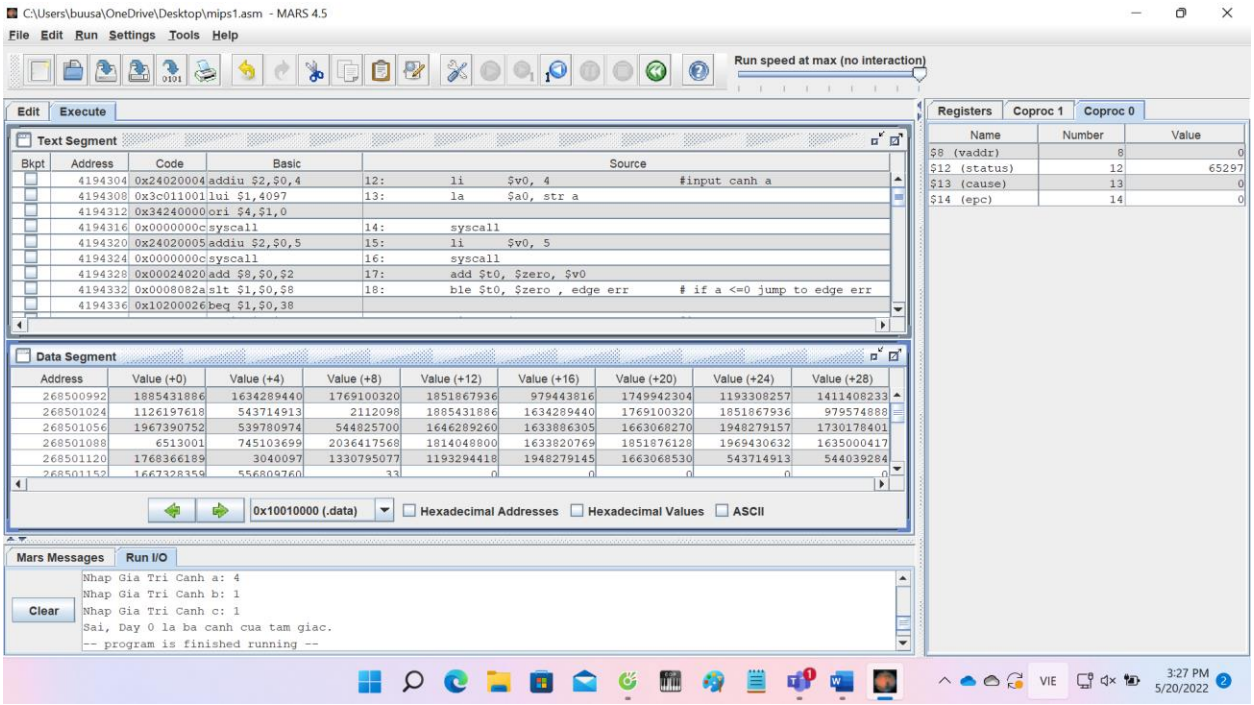
```
        syscall
        add $t1, $zero, $v0
        ble $t1, $zero , edge_err  # if b <=0 jump to edge_err

        li     $v0, 4               #input canh c
        la     $a0, str_c
        syscall
        li      $v0, 5
        syscall
        add $t2, $zero, $v0
        ble $t2, $zero , edge_err  # if c <=0 jump to edge_err
        #if a+b <c jump toi no_triagle
        add $a1,$t0,$t1
        blt $a1,$t2,no_triagle
        #if b+c <a jump toi no_triagle
        add $a1,$t1,$t2
        blt $a1,$t0,no_triagle
        #if c+a <b jump to no_triagle
        add $a1,$t2,$t0
        blt $a1,$t1,no_triagle

        j yes_triagle # neu thoa man thi jump toi yes_triagle
no_triagle:          #in ra thong bao ko phai 3 canh tam giac
        li $v0,4
        la $a0,no
        syscall
        j end
yes_triagle:         #in ra thong bao day la 3 canh cua tam giac
        li $v0,4
        la $a0,yes
        syscall
        j end
edge_err:            #in ra thong bao canh phai >0
        li $v0, 4
        la $a0,err
        syscall
        j end
end:                 #ket thuc chuong trinh
```

```
        li $v0,10
        syscall
```

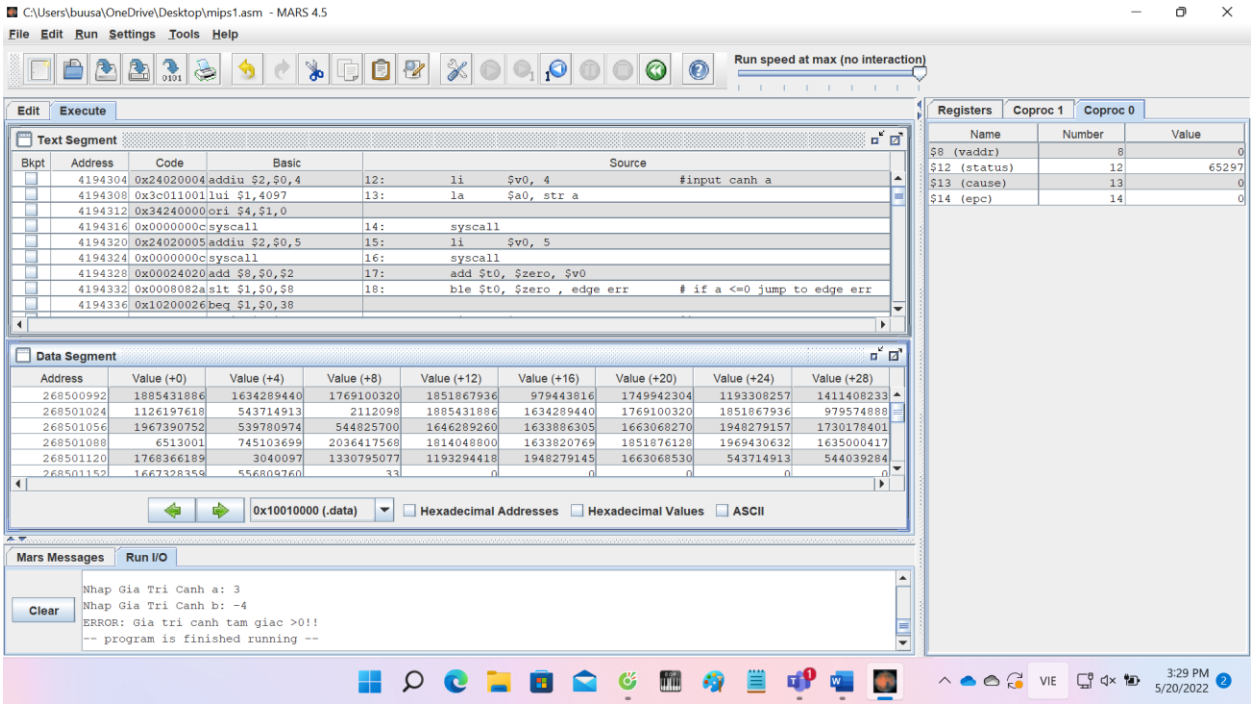
TH1: Thỏa mãn điều kiện 3 cạnh tam giác:



TH2: Không thỏa mãn điều kiện 3 cạnh tam giác:



TH2: Không thỏa mãn điều kiện 3 cạnh tam giác:

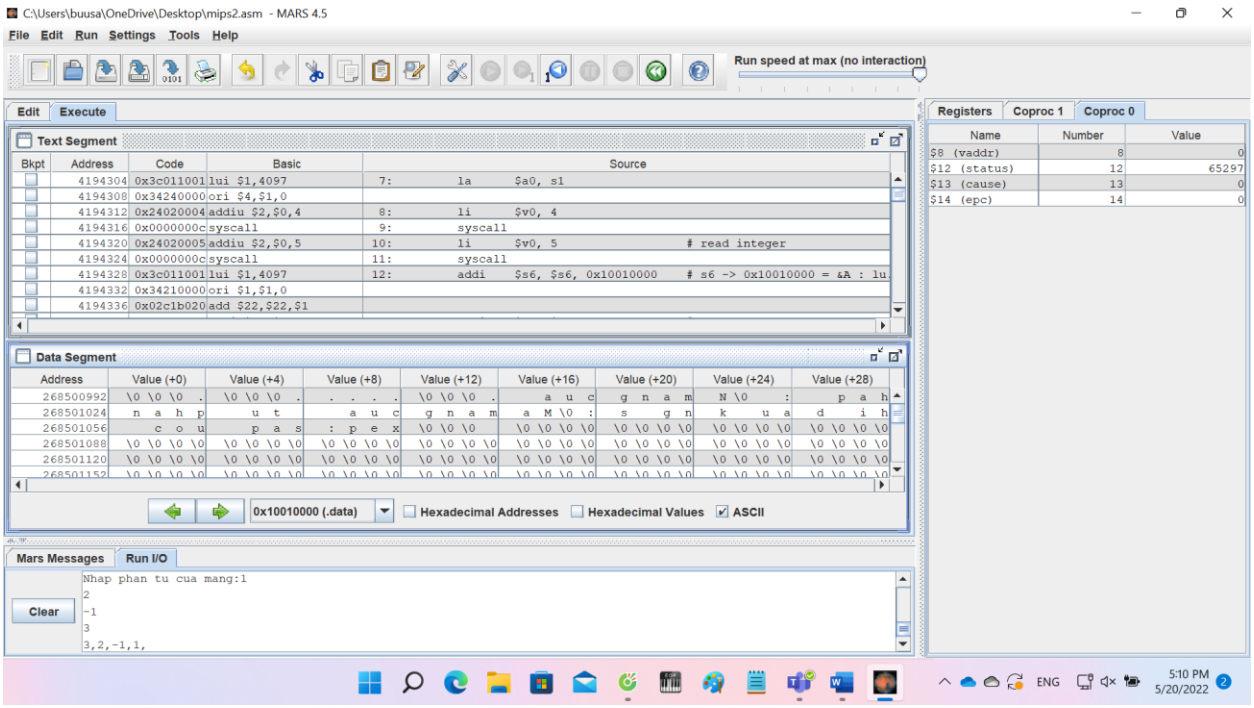


Assignment 2

```
.data
	str1:	.asciiz	"Nhap so phan tu cua mang: "
	str2:	.asciiz	"Nhap phan tu cua mang:"
	str3:	.asciiz	"Mang sau khi duoc sap xep: "
.text
	la	$a0, s1
	li	$v0, 4
	syscall
	li	$v0, 5			# read integer
	syscall
	addi	$s6, $s6, 0x10010000	# s6 -> 0x10010000 = &A : luu dia chi cua bo nho
	addi	$s5, $s6, 0		# s5 -> 0x10010000 = &A : luu dia chi cua bo nho
	addi	$s0, $v0, 0		# s0 = N
	addi	$t1, $zero, 0		# i = 0
	mul	$s1, $s0, 4		# s1 = 4 * N
	sub	$sp, $sp, $s1		# Khoi tao bo nho stack 4*n byte
	la	$a0, s2
	li	$v0, 4
	syscall
loop_scan:
	slt	$t0, $t1, $s0		# if (i < N)
	beq	$t0, $zero, end_loop_scan

	li	$v0, 5			# read integer
	syscall
	sw	$v0, 0($s5)		# s5[i] = v0

	addi	$t1, $t1, 1		# i++
	addi	$s5, $s5, 4		# s5 -> &A[i]

	j	loop_scan
end_loop_scan:
main:
	addi	$a0, $s6, 0		# a0 = &A[0]
	addi	$a1, $s0, 0		# s1 = N
	jal	sort
```

```
        addi   $t1, $zero, 0              # i = 0
        addi   $s5, $s6, 0          # s5 = s6 -> &A[0]
loop_print:
        slt    $t0, $t1, $s0              # if (i < N)
        beq    $t0, $zero, end_loop_print
        li     $v0, 1              # print integer
        lw     $a0, 0($s5)   # a0 = A[i]
        syscall
        li     $v0, 11       # print character
        addi   $a0, $zero, 44
        syscall
        addi   $t1, $t1, 1   # i++
        addi   $s5, $s5, 4   # s5 -> &A[i]
        j      loop_print
end_loop_print:
end_main:
        j      exit
swap:
        sw     $s0,0($t9)           # A[j] = A[i] truoc day
        sw     $s1,0($t8)           # A[i] = A[j] truoc day
        jr     $ra                  # Quay lai va tiep tuc vong lap
sort:
        addi   $sp, $sp, -8         # Khoi tao 2 vung nho 4 byte
        sw     $ra, 4($sp)          # push $ra -> stack
        sw     $s0, 0($sp)          # push        $s0 -> stack

        addi   $t1, $zero, 0        # i = 0
        addi   $t2, $zero, 0        # j = 0
        addi   $t8, $a0, 0          # v0 -> &A[0]
loop_i:
        slt    $t0, $t1, $a1        # if (i < n)
        beq    $t0, $zero, end_loop_i

        lw     $s0, 0($t8)          # s0 = A[i]
        bltz   $s0, continue_i            # if (A[i] < 0)

        addi   $t2, $t1, 1          # j = i+1
```

```
        addi   $t9, $t8, 4          # v1 -> A[i+1]
loop_j:
        slt    $t0, $t2, $a1        #if (j < n)
        beq    $t0, $zero, end_loop_j

        lw     $s1, 0($t9)          # s1 = A[j]
        bltz   $s1, continue_j            # if (s1 < 0) -> continue_j

        slt     $t0, $s1, $s0             # if (A[i] > A[j]) -> swap
        bne    $t0, $zero, continue_j
        jal    swap
continue_j:
        addi   $t2, $t2, 1          # j++
        addi   $t9, $t9, 4          # t9 -> &A[j]
        lw     $s0, 0($t8)          # s0 = A[i]
        j      loop_j
end_loop_j:
continue_i:
        addi   $t1, $t1, 1          # i++
        addi   $t8, $t8, 4          # t8 -> &A[i]
        j      loop_i
end_loop_i:
        lw     $s0, 0($sp)
        lw     $ra, 4($sp)
        jr     $ra
end_sort:
exit:
```



ASSIGNMENT 3:

```
int n=0;
for(int i=0;i<strlen(arr);i++)
{
      if(B[i]==0)
        for(int j=0;j<strlen(arr);j++)
          if((B[j]==0) &&(arr[i]==arr[j]) && (j!=i))
          {
            B[j]=1;
     n--;
          }
        n++;
        B[i]=1;
}
```

```
.data
        string: .space   50          #Khai bao chuoi co do dai 50
        Message1:    .asciiz "Nhap xau:" #khai bao chuoi ky tu "Nhap xau:" voi nhan Message1
        Message2:    .asciiz "So ky tu khac nhau:"          #tuong tu
        arr:       .word 0:100
.text
main:
get_string:
        li         $v0, 54           #gan thanh ghi v0 = 54 : hien hop thoai nhap chuoi
        la         $a0, Message1  #load dia chi bien Massage1 vao thanh ghi a0
        la         $a1, string       #load dia chi bien string vao thanh ghi a1
        la         $a2, 100          #khai bao a2 = 100
        syscall                      #goi ham syscall voi v0 = 54 -> hien hop thoai nhap chuoi
get_length:
        la         $a0,string# $a0=address(string[0])
        la $a1 ,arr
        add        $v1,$zero,$zero# $v1= length =0
        add        $t0,$zero,$zero# $t0=i=0
        add        $s0,$zero,$zero# $s0=j=0
        add        $t6,$zero,$zero# $t6=n=0
check_char:
        add        $t1,$a0,$t0      # $t1=$a0+$t0
                                    #=address(string[i])
        lb         $t3, 0($t1)       #t3 = string[i]
        beq        $t3, $zero,end_get_length
        addi       $v1,$v1,1         #v1 = length++
        addi       $t0,$t0,1         #t0 = i++
        j check_char                 #jump to check_char
```

```
end_get_length:
        addi    $v1,$v1, -1

        add     $t0,$zero,$zero# $t0=i=0
loop:

        add     $t1,$a1,$t0     # $t1=$a0+$t0
                                #=address(arr[i])

        lb      $t3, 0($t1)     #t3 = arr[i]
        beqz    $t3,loop2
        addi    $t0,$t0,1       #t0 = i++
blt $s0,$v1,loop
loop:

        add     $t4,$a1,$s0     # $t1=$a0+$s0
                                #=address(arr[j])
        lb      $t5, 0($t4)     #t3 = arr[j]
        addi    $s0,$s0,1       #t0 = j++
        beqz
blt $s0,$v1,loop



    return n;
}
```