

BÁO CÁO MINI-PROJECT
Lương Thị Tâm – 20194663

32	20194663	Lương Thị Tâm	6	3	4
----	----------	---------------	---	---	---

Bài 1(A-6):Nhập 3 số nguyên a, b, c, kiểm tra đây có phải là 3 cạnh của một tam giác không.

*Phân tích cách thực hiện:

Điều kiện để 3 số là cạnh của 1 tam giác là tổng hai số bất kì luôn lớn hơn số còn lại.

Thực hiện nhập vào lần lượt 3 số bằng cách gọi hàm $\$v0 = 5$

Sau đó kiểm tra điều kiện trên bằng cách sử dụng lệnh cộng add để tính tổng, thực hiện kiểm tra điều kiện bằng cặp lệnh slt, beq

*Kết quả thực hiện:

Clear	Nhap: -1 Nhap sai. So nhap vao phai la so duong! Nhap: 1 Nhap: 2 Nhap: 3 Ba so vua nhap khong la ba canh cua mot tam giac! -- program is finished running --
	Nhap: 3 Nhap: 4 Nhap: 5 Ba so vua nhap la ba canh cua mot tam giac! -- program is finished running --

*Mã nguồn:

```
1 .data
2 message1: .asciiiz "Nhap: "
3 message2: .asciiiz "Ba so vua nhap la ba canh cua mot tam giac!"
4 message3: .asciiiz "Ba so vua nhap khong la ba canh cua mot tam giac!"
5 message4: .asciiiz "Nhap sai.So nhap vao phai la so duong!\n"
6 .text
7 #Nhap vao ba so
8 nhap: li $v0, 4
9     la $a0, message1
10    syscall
11
12    li $v0, 5 # nhap vao a
13    syscall
14    addi $t1, $v0, 0 # $t1 = a
15    bltz $t1, check
16
17    li $v0, 4
18    la $a0, message1
19    syscall
20
21    li $v0, 5 # nhap vao b
22    syscall
23    addi $t2, $v0, 0 # $t2 = b
24    bltz $t2, check
25
26    li $v0, 4
27    la $a0, message1
28    syscall
29
30    li $v0, 5 #nhap vao c
31    syscall
32    addi $t3, $v0, 0 # $t3 = c
33    bltz $t3, check
34    nop
35    j main
36    nop
37    # kiem tra dieu kien nhap
```

```
37    # kiem tra dieu kien nhap
38 check: li $v0, 4
39     la $a0, message4
40     syscall
41     j nhap # quay lai de nhap
42     #kiem tra dieu kien tam giac
43 main: add $s0, $t1, $t2
44     slt $t4, $t3, $s0 # a + b > c ?
45     beq $t4, $zero, return1 # a + b <= c, return1
46     nop
47
48     add $s0, $t2, $t3
49     slt $t4, $t1, $s0 # b + c > a?
50     beq $t4, $zero, return1 # b+c<=a ,return1
51     nop
52
53     add $s0, $t1, $t3
54     slt $t4, $t2, $s0 # a+c>b?
55     beq $t4, $zero, return1 #a+c<=b, return1
56     nop
57     j return2
58
59 return1: li $v0, 4
60     la $a0, message3
61     syscall
62
63     li $v0, 10
64     syscall
65
66 return2: li $v0, 4
67     la $a0, message2
68     syscall
69
70     li $v0, 10
71     syscall
```

Bài 2(B-3):Nhập mảng số nguyên từ bàn phím. In ra màn hình tổng các phần tử lẻ và chia hết cho 5 trong mảng.

*Phân tích cách thực hiện:

*Ý nghĩa chương trình con:

*Kết quả thực hiện:

```
Nhap so phan tu cua mang: 5
5
7
9
10
15
Clear
Cac phan tu thoa man la:
5
15
-- program is finished running (dropped off bottom) --
```

*Mã nguồn:

```
1 .data
2 arr: .space 200
3 str: .asciiz "Nhap so phan tu cua mang: "
4 str1: .asciiz "Cac phan tu thoa man la: \n"
5 str2: .asciiz "\n"
6 .text
7 la $s0, arr
8 la $a0, str
9 addi $v0, $0, 4
10 syscall
11 addi $v0, $0, 5
12 syscall
13 add $t1, $v0, $v0
14 add $t1, $t1, $t1 #vi tri cuoi cua mang
15 addi $t0, $0, 0 #i=0
16 for:
17 bge $t0, $t1, end_for #if i >= vi tri cuoi, exit
18 add $s1, $s0, $t0 #arr[i]
19 addi $v0, $0, 5 #doc phan tu vao mang
20 syscall
21 sw $v0, 0($s1) # lay gia tri phan tu vua doc vao
22 count:
23 addi $t0, $t0, 4 #i++
24 j for #
25 end_for:
26 addi $t0, $0, 0 #i=0
27
28 la $a0, str1
29 addi $v0, $0, 4
30 syscall
31
32 for1:
33 bge $t0, $t1, end_for1 #if i >= vi tri cuoi, exit
34 add $s1, $s0, $t0 #arr[i]
35 lw $v0, 0($s1) #arr[i]
36 addi $v1, $0, 5
37 div $v0, $v1
38 mfhi $v1
39 bne $v1, $0, count1 #neu arr[i] chia het cho 5, tiep tuc, neu khong nhay den count1
40 addi $v1, $0, 2
41 div $v0, $v1
42 mfhi $v1
43 beq $v1, $0, count1 #neu arr[i] chia het cho 2, continue
44 addi $a0, $v0, 0
45 addi $v0, $0, 1 #in ra phan tu thoa man dk arr[i]
46 syscall
47 la $a0, str2
48 addi $v0, $0, 4
49 syscall
50 count1:
51 addi $t0, $t0, 4
52 j for1
53 end_for1:
```

Bài 3(C-4):Nhập vào 2 chuỗi ký tự s1 và s2, in ra màn hình các ký tự khác nhau xuất hiện cả trong 2 chuỗi.

*Phân tích cách thực hiện:

*Ý nghĩa chương trình con:

*Kết quả thực hiện:

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	p a h N	o a v	u a x	t \0 :	\0 \n m a	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	t \0 \0 \0	\0 \n u h	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010140	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010160	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Mars MessagesRun I/O

Nhap vao xau: tam
Nhap vao xau: thu
t

-- program is finished running (dropped off bottom) --

Clear

*Mã nguồn:

```

1  .data
2  message: .asciiz "Nhap vao xau: "
3  string1: .space 100
4  string2: .space 100
5  .text
6      li $v0, 4
7      la $a0, message
8      syscall
9
10     li $v0, 8 #nhap vao xau thu nhat
11     la $a0, string1
12     la $a1, 100
13     syscall
14     add $s6, $a0, $zero # lay dia chi xau thu nhat str1[0]
15
16     li $v0, 4
17     la $a0, message
18     syscall
19
20     li $v0, 8 #nhap vao xau thu hai
21     la $a0, string2
22     la $a1, 100
23     syscall
24
25     add $s7, $a0, $zero # lay dia chi xau thu hai
26
27     li $t0, 0 # bien chay cho vong while 1
28     li $t5, 0 # bien luu ki tu da in ra
29 loop: add $s4, $s6, $t0
30     lb $t1, 0($s4) #load ki tu vao $t1
31     beq $t1, $zero, end_loop # load het xau, exit
32     nop
33     li $t3, 0 #j = 0
34
35 loop2: add $s5, $s7, $t3
36     lb $s1, 0($s5) # load ki tu vao $s1
37     beq $s1, $zero, end_loop2 #load het xau, exit
38     nop
39     beq $t1, $s1, return # neu str1[i] = str[j], nhay den return
40     addi $t3, $t3, 1 # j = j + 1
41     j loop2
42 end_loop2: addi $t0, $t0, 1 # i = i + 1
43     j loop
44
45 return: beq $t1, $t5, end_loop2 # neu giong ky tu da in ra, bo qua va quay lai vong lap
46     li $v0, 11
47     addi $a0, $t1, 0
48     syscall
49     add $t5, $t1, $zero # luu lai ki tu vua in ra
50     addi $t3, $t3, 1 # j = j +1
51     j loop2
52 end_loop:
53
54
55
56

```

Line: 19 Column: 2 ☒ Show Line Numbers

Mars Messages Run I/O

Clear