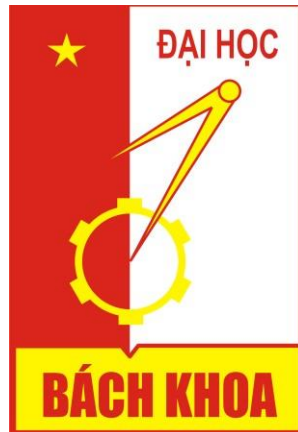


ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường công nghệ Thông tin & Truyền thông



IT3280

Thực hành Kiến trúc máy tính

BÁO CÁO MINI-PROJECT

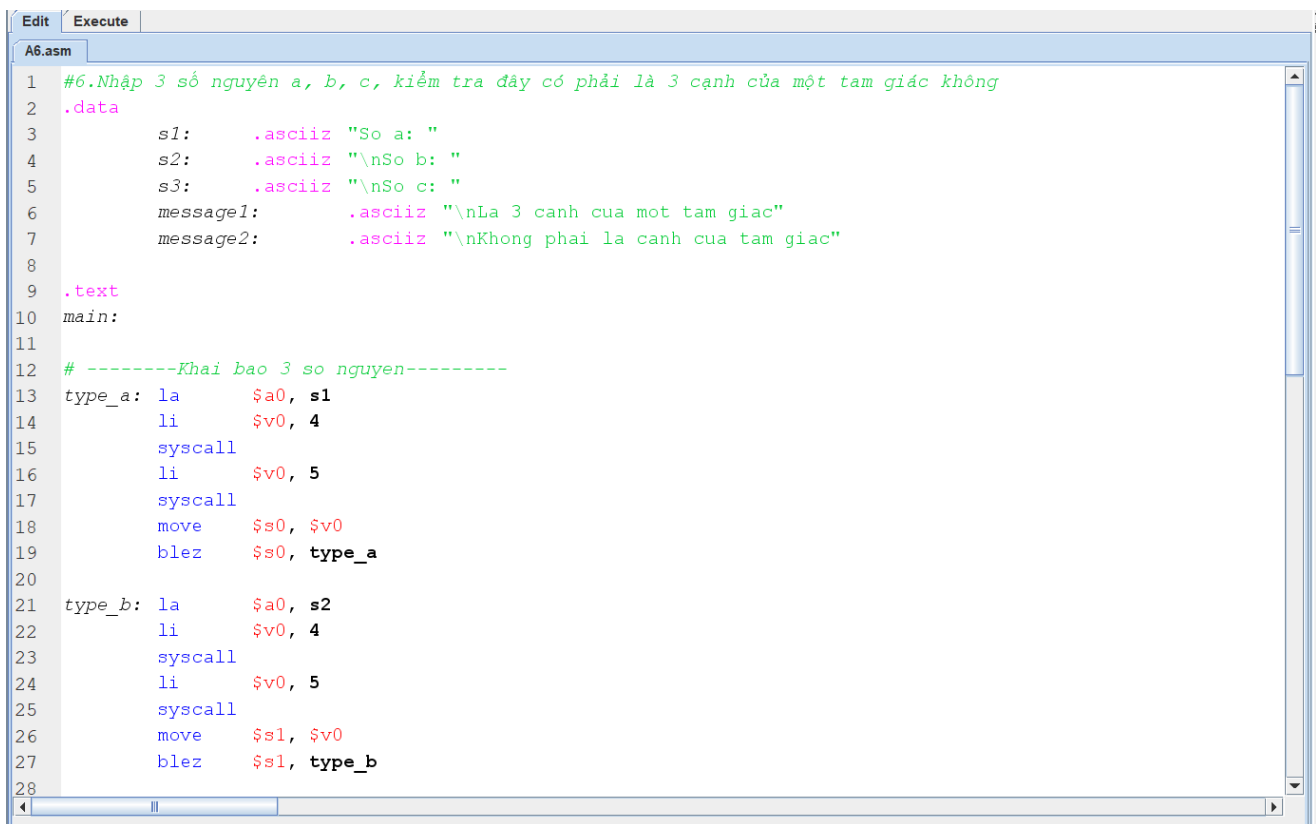
Giảng viên hướng dẫn	:	ThS. Lê Bá Vui
Mã lớp	:	130938
Sinh viên	:	Mai Đào Tuấn Thành
MSSV	:	20194675

A. Số nguyên

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

Phân tích: Ta nhập 3 số nguyên từ bàn phím. Đầu tiên kiểm tra xem các số có lớn hơn 0 hay không. Sau đó thực hiện nếu tổng của 2 số lớn hơn số còn lại thì là tam giác.

Chương trình:



```
1  #6.Nhập 3 số nguyên a, b, c, kiểm tra đây có phải là 3 cạnh của một tam giác không
2  .data
3      s1:      .asciiz "So a: "
4      s2:      .asciiz "\nSo b: "
5      s3:      .asciiz "\nSo c: "
6      message1: .asciiz "\nLa 3 cạnh của mot tam giac"
7      message2: .asciiz "\nKhong phai la cạnh của tam giac"
8
9  .text
10 main:
11
12 # -----Khai bao 3 so nguyen-----
13 type_a: la      $a0, s1
14         li      $v0, 4
15         syscall
16         li      $v0, 5
17         syscall
18         move    $s0, $v0
19         blez    $s0, type_a
20
21 type_b: la      $a0, s2
22         li      $v0, 4
23         syscall
24         li      $v0, 5
25         syscall
26         move    $s1, $v0
27         blez    $s1, type_b
28
```

```

A6.asm
29 type_c: la    $a0, s3
30         li    $v0, 4
31         syscall
32         li    $v0, 5
33         syscall
34         move  $s2, $v0
35         blez  $s2, type_c
36         #-----
37 check:
38
39         add    $t0, $s0, $s1          # t0 = a + b
40         ble    $t0, $s2, false        # if a + b <= c return false
41         nop
42         # OR
43         add    $t0, $s1, $s2          # t0 = b + c
44         ble    $t0, $s0, false        # if b + c <= a return false
45         nop
46         # OR
47         add    $t0, $s2, $s0          # t0 = a + b
48         ble    $t0, $s1, false        # if a + c <= b return false
49         nop
50         j      end_main

```

```

51 false:
52         la    $a0, message2          # neu khong phai la tam giac thi in ra message 2
53         li    $v0, 4
54         syscall
55         j      exit
56 end_main:
57         la    $a0, message1          # neu la tam giac thi in message 1
58         li    $v0, 4
59         syscall
60 exit:
61
62
63
64

```

Line: 57 Column: 54 [x] Show Line Numbers

Kết quả thực hiện:

Mars Messages	Run I/O
<div>Clear</div>	So a: 3
	So b: 4
	So c: 5
	La 3 canh cua mot tam giac
	-- program is finished running (dropped off bottom) --

Mars Messages Run I/O

So a: 1

So b: 1

So c: 2

Khong phai la canh cua tam giac

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

Clear

B. Mạng

Bài 2: Nhập mảng số nguyên từ bàn phím. Sắp xếp các phần tử có giá trị dương giảm dần.

Ý tưởng: Lưu các số nguyên vào trong mảng, sử dụng vòng lặp để duyệt tìm, nếu thấy số âm thì bỏ qua và quay lại vòng lặp, nếu thấy số dương thì so sánh và đổi chỗ theo yêu cầu.

Chương trình:

Edit Execute

B_2.asm

```

1  #B2.Nhập mảng số nguyên từ bàn phím. Sắp xếp các phần tử có giá trị dương giảm dần
2  .data
3      s1:      .asciiz "Nhap so phan tu cua mang: "
4      s2:      .asciiz "\nNhap phan tu cua mang:\n"
5      s3:      .asciiz "\nMang sau khi duoc sap xep: "
6  .text
7      la      $a0, s1
8      li      $v0, 4
9      syscall
10
11     li      $v0, 5          # read integer
12     syscall
13
14     addi    $s6, $s6, 0x10010000 # s6 -> 0x10010000 = &A : luu dia chi cua bo nho
15     addi    $s5, $s6, 0        # s5 -> 0x10010000 = &A : luu dia chi cua bo nho
16
17     addi    $s0, $v0, 0        # s0 = N
18     addi    $t1, $zero, 0      # i = 0
19
20     mul     $s1, $s0, 4        # s1 = 4 * N
21     sub     $sp, $sp, $s1      # Khoi tao bo nho stack 4*n byte
22
23     la      $a0, s2
24     li      $v0, 4
25     syscall
26 loop_scan:
27     slt     $t0, $t1, $s0      # if (i < N)
28     beq     $t0, $zero, end loop_scan

```

```
B_2.asm
29
30     li      $v0, 5                # read integer
31     syscall
32     sw      $v0, 0($s5)          # s5[i] = v0
33
34     addi    $t1, $t1, 1           # i++
35     addi    $s5, $s5, 4           # s5 -> &A[i]
36
37     j       loop_scan
38 end_loop_scan:
39
40 main:
41     addi    $a0, $s6, 0           # a0 = &A[0]
42     addi    $a1, $s0, 0           # s1 = N
43     jal     sort
44
45     addi    $t1, $zero, 0         # i = 0
46     addi    $s5, $s6, 0           # s5 = s6 -> &A[0]
47 loop_print:
48     slt     $t0, $t1, $s0         # if (i < N)
49     beq     $t0, $zero, end_loop_print
50
51     li      $v0, 1                # print integer
52     lw      $a0, 0($s5)           # a0 = A[i]
53     syscall
54
55     li      $v0, 11               # print character
56     addi    $a0, $zero, 44
57     syscall
58
59     addi    $t1, $t1, 1           # i++
60     addi    $s5, $s5, 4           # s5 -> &A[i]
61
62     j       loop_print
63 end_loop_print:
64 end_main:
65     j       exit
66 swap:
67     sw      $s0, 0($t9)           # A[j] = A[i] truoc day
68     sw      $s1, 0($t8)           # A[i] = A[j] truoc day
69     jr      $ra                   # Quay lai va tiep tuc vong lap
70 sort:
71     addi    $sp, $sp, -8          # Khoi tao 2 vung nho 4 byte
72     sw      $ra, 4($sp)           # push $ra -> stack
73     sw      $s0, 0($sp)           # push $s0 -> stack
74
75     addi    $t1, $zero, 0         # i = 0
76     addi    $t2, $zero, 0         # j = 0
77     addi    $t8, $a0, 0           # v0 -> &A[0]
78 loop_i:
79     slt     $t0, $t1, $a1         # if (i < n)
80     beq     $t0, $zero, end_loop_i
81
82     lw      $s0, 0($t8)           # s0 = A[i]
83     bltz    $s0, continue_i       # if (A[i] < 0)
```

```

85      addi    $t2, $t1, 1          # j = i+1
86      addi    $t9, $t8, 4          # v1 -> A[i+1]
87  loop_j:
88      slt     $t0, $t2, $a1        #if (j < n)
89      beq     $t0, $zero, end_loop_j
90
91      lw      $s1, 0($t9)          # s1 = A[j]
92      bltz    $s1, continue_j      # if (s1 < 0) -> continue_j
93
94      slt     $t0, $s1, $s0        # if (A[i] > A[j]) -> swap
95      bne     $t0, $zero, continue_j
96      jal     swap
97  continue_j:
98      addi    $t2, $t2, 1          # j++
99      addi    $t9, $t9, 4          # t9 -> &A[j]
100     lw      $s0, 0($t8)          # s0 = A[i]
101     j        loop_j
102 end_loop_j:
103 continue_i:
104     addi    $t1, $t1, 1          # i++
105     addi    $t8, $t8, 4          # t8 -> &A[i]
106     j        loop_i
107 end_loop_i:
108     lw      $s0, 0($sp)
109     lw      $ra, 4($sp)
110     jr      $ra
111 end_sort:
112 exit:

```

Kết quả:

Mars Messages Run I/O

```

Nhap so phan tu cua mang: 8

Nhap phan tu cua mang:
-1
-2
3
-5
1
7
-8
9
-1,-2,9,-5,7,3,-8,1,
-- program is finished running (dropped off bottom) --

```

Clear

C. Xâu ký tự

Bài 1. Nhập vào xâu ký tự. In ra màn hình từ ngắn nhất có trong xâu.

Ý tưởng: Nhập vào một xâu ký tự, thực hiện vòng lặp nếu gặp các ký tự khác dải ký tự từ A-Z và a-z thì ta sẽ tính là 1 từ mới. Kiểm tra độ dài của từ nếu thấy từ nào có độ dài ngắn hơn thì giữ lại.

Chương trình:

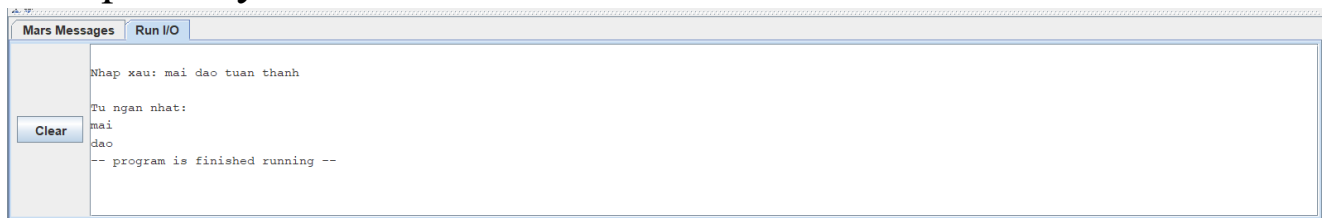
```
C1.asm
1  #1_C.Nhập vào xâu ký tự. In ra màn hình từ ngắn nhất có trong xâu.
2
3
4  .data
5      xau: .space 32          # Lưu xâu đã nhập
6      Dodai: .space 100
7      msg: .asciiz "\nNhập xâu: "
8      message: .asciiz "\nTu ngắn nhất: "
9      enter: .asciiz "\n"
10
11 .text
12 start: li $v0, 4
13        la $a0, msg
14        syscall
15
16        li $v0, 8
17        la $a0, Dodai
18        la $a1, 100
19        syscall
20
21        li $s0, 0            # i = 0
22        li $s2, 0            # j = 0
23        li $s4, 1111         # độ dài ngắn nhất
24        li $s6, 0            # k = 0
25        la $a2, xau          # load địa chỉ xâu
26
27 main:  jal check            # vòng lặp qua chuỗi đầu vào
28
29
30
31
32
33        la $a1, Dodai
34        j printWord
35
36 exit:  li $v0, 10
37        syscall
38
39 check:
40        add $t4, $s0, $a0     # address of A[i] in $t4
41        lb $s1, 0($t4)        # load value of A[i]
42
43        slti $t1, $s1, 65     # if ascii code is less than 48
44        bne $t1, $zero, checkLength
45
46        slti $t1, $s1, 91     # if ascii code is greater than 90
47
48        slti $t2, $s1, 97     # if ascii code is less than 97
49        slt $t3, $t1, $t2
50        bne $t3, $zero, checkLength
51
```

```

52      slti    $t1, $s1, 123                # if ascii character is greater than 122
53      beq     $t1, $zero, checkLength
54
55      addi    $s0, $s0, 1                  # i = i + 1
56      addi    $s2, $s2, 1                  # j = j + 1
57      j       check                        # go to check
58
59  checkLength:
60      slt     $t3, $s2, $s4                # so sánh với độ dài từ còn lại (j > min)
61      beqz    $t3, next                    # reset j và move đến bước tiếp theo
62      add     $s4, $zero, $s2              # ngược lại ta được min length mới
63  next:  beq     $s1, 10, done              # if A[i] = '\n' -> done
64      addi    $s0, $s0, 1                  # i = i + 1
65      li      $s2, 0                       # j = 0
66      j       check
67  done:  li      $s0, 0
68          li      $s2, 0
69          jr      $ra                      # return to main
70
71  printWord:
72      add     $t4, $s0, $a1                # address of A[i] in $t4
73      lb      $s1, 0($t4)                  # load value of A[i]
74
75      slti    $t1, $s1, 65                 # if ascii code is less than 65
76      bne     $t1, $zero, compare
77
78      slti    $t1, $s1, 91                 # if ascii code is greater than 90
79
80      slti    $t2, $s1, 97                 # if ascii code is less than 97
81      slt     $t3, $t1, $t2
82      bne     $t3, $zero, compare
83
84      slti    $t1, $s1, 123                # if ascii character is greater than 122
85      beq     $t1, $zero, compare
86
87      add     $t5, $s2, $a2                # address of xau[i] in $t5
88      sb      $s1, 0($t5)                  # store current character to xau[i]
89      addi    $s0, $s0, 1                  # i = i + 1
90      addi    $s2, $s2, 1                  # j = j + 1
91      j       printWord
92
93  compare:
94      beq     $s2, $s4, print              # if length of current word is equal to current max
95      j       reset                        # reset j and move
96
97  print:  li      $v0, 4
98          la      $a0, enter
99          syscall
100         li      $v0, 4
101         la      $a0, xau                  # word with max length
102         syscall
103
104  reset:  addi    $s0, $s0, 1                # i = i + 1
105         li      $s6, 0                    # k = 0
106
107  clear_xau:
108      beq     $s6, $s2, return              # if k = j end
109      add     $t4, $s6, $a2                # loop through xau string
110      sb      $0, 0($t4)                    # turn xau[k] into 0
111      addi    $s6, $s6, 1                  # k = k + 1
112      j       clear_xau                    # continue to clear
113
114  return: li      $s2, 0                    # j = 0
115         beq     $s1, 10, end_main          # if A[i] = '\n' -> done. Press enter to continue
116         j       printWord
117
118  end_main:
119      j       exit

```


Kết quả chạy:



Check: kiểm tra để tách lấy các từ
Checklength: kiểm tra độ dài xâu