

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA



MÔN HỌC: KIẾN TRÚC MÁY TÍNH (THỰC HÀNH) (CO2008)

Bài tập/Thực hành 3

CHƯƠNG 2 KIẾN TRÚC TẬP LỆNH MIPS: CÁC LỆNH ĐIỀU KHIỂN

LỚP THỰC HÀNH L03 – HỌC KỲ 212

Giảng viên hướng dẫn: Vũ Trọng Thiên

Sinh viên thực hiện

Phạm Duy Quang

Mã số sinh viên

2011899

Thành phố Hồ Chí Minh, tháng 03 năm 2022

Bài tập và Thực hành

Lập trình có cấu trúc.

Sinh viên chuyển các cấu trúc sau của ngôn ngữ C qua ngôn ngữ assembly.

Bài 1: Phát biểu IF-ELSE (1)

```
1 if( a % 2 == 1) { Print string: "Computer Science and Engineering, HCMUT"}  
2 else           { Print string: "Computer Architecture 2020"}
```

❖ Code:

```
.data  
str: .ascii "a = "  
str1: .ascii "Computer Science and Engineering, HCMUT"  
str2: .ascii "Computer Architecture 2020"  
.text  
    li $v0, 4  
    la $a0, str  
    syscall  
    li $t0, 2  
  
    li $v0, 5  
    syscall  
    move $t1, $v0  
  
    div $t1, $t0  
    mfhi $t6  
  
.globl main  
main:  
    beq $t6, 1, TrueCase  
    li $v0, 4  
    la $a0, str2  
    syscall
```

```
j Fin
```

```
TrueCase:
```

```
li $v0, 4
```

```
la $a0, str1
```

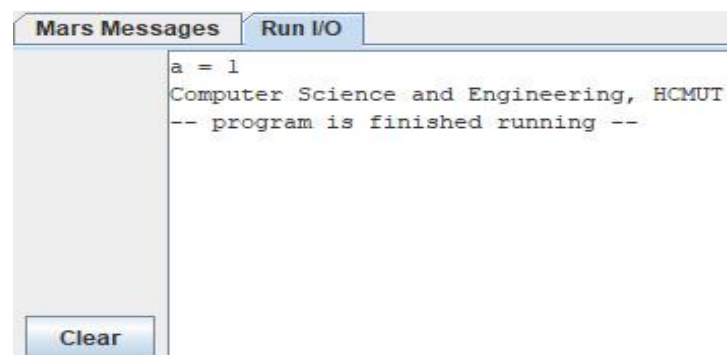
```
syscall
```

```
Fin:
```

```
li $v0, 10
```

```
syscall
```

❖ Run I/O:



Bài 2: Phát biểu IF-ELSE (2)

```
1 if( a < -5 or a >= 3 ) { a = b * c; }  
2 else { a = b + c; }
```

❖ Code:

```
.data  
str1: .asciiz "a = "  
str2: .asciiz "b = "  
str3: .asciiz "c = "  
str4: .asciiz "a = b * c = "  
str5: .asciiz "a = b + c = "  
.text  
li $s1, -5  
li $s2, 3
```

```
li $v0, 4
la $a0, str1
syscall
li $v0, 5
syscall
move $t0, $v0
```

```
li $v0, 4
la $a0, str2
syscall
li $v0, 5
syscall
move $t1, $v0
```

```
li $v0, 4
la $a0, str3
syscall
li $v0, 5
syscall
move $t2, $v0
```

```
.globl main
```

```
main:
slt $t3, $t0, $s1
sge $t4, $t0, $s2
bne $t3, $zero, TrueCase
bne $t4, $zero, TrueCase
add $t0, $t1, $t2
li $v0, 4
la $a0, str5
syscall
```

```

j Fin

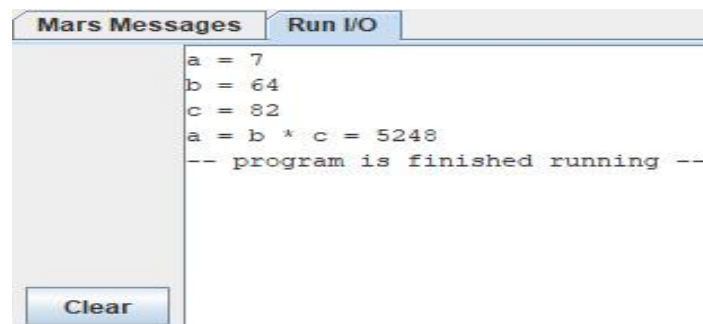
TrueCase:
mul $t0, $t1, $t2
li $v0, 4
la $a0, str4
syscall

Fin:
move $s0, $t0
li $v0, 1
la $a0, ($s0)
syscall

li $v0, 10
syscall

```

❖ Run I/O:



Bài 3: Phát biểu SWITCH-CASE

Hiện thực phát biểu switch-case bên dưới bằng hợp ngữ. Cho biết $b = 200$, $c = 4$. Giá trị input nhập từ người dùng. **Xuất ra giá trị của a ra màn hình (console).**

```

1 switch (input)
2 {
3     case 1: a = b + c; break;
4     case 2: a = b - c; break;
5     case 3: a = b x c; break;
6     case 4: a = b / c; break;
7     default: NOP; // No-Operation; a = 0
8             break;
9 }

```

❖ **Code:**

```
.data
str1: .asciiz "input = "
str2: .asciiz "a = "
.text
    li $s1, 200
    li $s2, 4

    li $v0, 4
    la $a0, str1
    syscall
    li $v0, 5
    syscall
    move $t0, $v0

.globl main
main:
    beq $t0, 1, Case1
    beq $t0, 2, Case2
    beq $t0, 3, Case3
    beq $t0, 4, Case4

    li $s0, 0
    j Fin

Case1:
    add $s0, $s1, $s2
    j Fin

Case2:
    sub $s0, $s1, $s2
    j Fin
```

Case3:

```
mul $s0, $s1, $s2
```

```
j Fin
```

Case4:

```
div $s1, $s2
```

```
mflo $s0
```

```
j Fin
```

Fin:

```
li $v0, 4
```

```
la $a0, str2
```

```
syscall
```

```
li $v0, 1
```

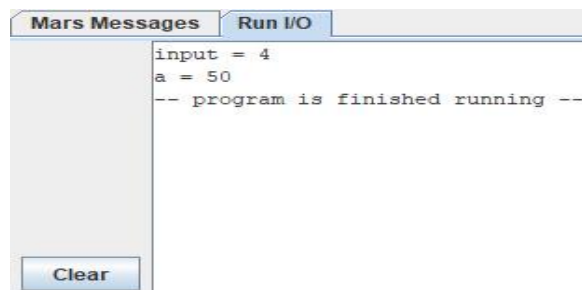
```
la $a0, ($s0)
```

```
syscall
```

```
li $v0, 10
```

```
syscall
```

❖ Run I/O:



Bài 4: Vòng lặp FOR - xác định chuỗi Fibonacci bằng vòng lặp. Nhập vào n (nguyên dương), xuất ra số Fibonacci Fn.

```

1  if      (n == 0) {return 0;}
2  else if (n == 1) {return 1;}
3  else{
4      f0= 0; f1 = 1;
5      for ( i = 2; i <= n; i++){
6          fn = fn-1 + fn-2;
7      }
8  }
9  return fn;

```

Note: sinh viên có thể làm theo cách riêng để tìm ra số Fibonacci Fn.

Dãy số Fibonacci: http://en.wikipedia.org/wiki/Fibonacci_number

F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉
0	1	1	2	3	5	8	13	21	34
F ₁₀	F ₁₁	F ₁₂	F ₁₃	F ₁₄	F ₁₅	F ₁₆	F ₁₇	F ₁₈	F ₁₉
55	89	144	233	377	610	987	1597	2584	4181

❖ **Code:**

```

.text
main:
# Prompt user to input non-negative number
la $a0,prompt
li $v0,4
syscall

li $v0,5 #Read the number(n)
syscall

move $t2,$v0 # n to $t2

# Call function to get fibonnacci #n
move $a0,$t2
move $v0,$t2
jal fib #call fib (n)
move $t3,$v0 #result is in $t3

# Output message and n

```



```

la $a0,result #Print F_
li $v0,4
syscall

move $a0,$t2 #Print n
li $v0,1
syscall

la $a0,result2 #Print =
li $v0,4
syscall

move $a0,$t3 #Print the answer
li $v0,1
syscall

la $a0,endl #Print '\n'
li $v0,4
syscall

# End program
li $v0,10
syscall

fib:
# Compute and return fibonacci number
beqz $a0,zero #if n=0 return 0
beq $a0,1,one #if n=1 return 1

#Calling fib(n-1)
sub $sp,$sp,4 #storing return address on stack
sw $ra,0($sp)

```

```

sub $a0,$a0,1  #n-1
jal fib    #fib(n-1)
add $a0,$a0,1

lw $ra,0($sp)  #restoring return address from stack
add $sp,$sp,4

sub $sp,$sp,4  #Push return value to stack
sw $v0,0($sp)
#Calling fib(n-2)
sub $sp,$sp,4  #storing return address on stack
sw $ra,0($sp)

sub $a0,$a0,2  #n-2
jal fib    #fib(n-2)
add $a0,$a0,2

lw $ra,0($sp)  #restoring return address from stack
add $sp,$sp,4
#-----
lw $s7,0($sp)  #Pop return value from stack
add $sp,$sp,4

add $v0,$v0,$s7 # f(n - 2)+fib(n-1)
jr $ra # decrement/next in stack

zero:
li $v0,0
jr $ra
one:

```

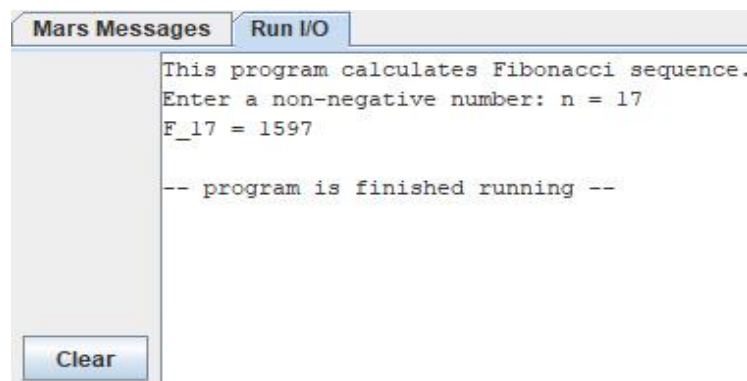
```

li $v0,1
jr $ra

.data
prompt: .ascii "This program calculates Fibonacci sequence.\nEnter a non-negative
number: n = "
result: .ascii "F_"
result2: .ascii " = "
endl: .ascii "\n"

```

❖ Run I/O:



Bài 5: Vòng lặp WHILE

Xác định vị trí chữ 'e' đầu tiên trong chuỗi "Computer Architecture CSE-HCMUT".

```

1 i = 0;
2 while( charArray[i] != 'e' && charArray[i] != '\0'){
3     i++;
4 }

```

Xuất ra giá trị index của ký tự 'e'. Nếu không tìm thấy thì xuất ra -1.

❖ Code:

```

.data
string: .ascii "Computer Architecture CSE-HCMUT"
size:      .word 30

.text
main:
    li $t0, 0

```

```
la $t1, string
```

```
while:
```

```
lb $t2, ($t1)
```

```
beq $t2, '\0', exit2
```

```
beq $t2, 'e', exit1
```

```
add $t1, $t1, 1
```

```
add $t0, $t0, 1
```

```
j while
```

```
exit1:
```

```
move $s0, $t0
```

```
li $v0, 1
```

```
la $a0, ($s0)
```

```
syscall
```

```
j fin
```

```
exit2:
```

```
li $s0, -1
```

```
li $v0, 1
```

```
la $a0, ($s0)
```

```
syscall
```

```
j fin
```

```
fin:
```

```
li $v0, 10
```

```
syscall
```

❖ Run I/O:

