

```
#####
#The Fibonacci numbers are the numbers in the following integer sequence.
# 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...
# F(0)=0 , F(1) =1
#int fibonacci(int n) {
# int f0=0;
# int f1=1;
# int x,i;
# if(n==0) return f0;
#
# i=1;
# while(i<n){
# x=f0+f1;
# f0=f1;
# f1=x;
# i++;
# }
# return f1;
#}
#####
.data
n: .word 5
sonuc: .space 4
```

```
.text
.globl main
```

```
main: la $t0,n
lw $t0,($t0)
addi $sp,$sp,-4
sw $t0, 0($sp) # n is our argument
jal fib # Call the fib procedure
addi $sp, $sp, 4 # Pop off the argument
la $s0,sonuc
sw $v0,0($s0) # result is in $v0
li $v0, 10 # code for program end
syscall

fib: addi $sp, $sp, -20
sw $ra, 16($sp) # push return address into the stack
sw $zero, 12($sp) # local variable f0
addi $t0,$zero,1
sw $t0,8($sp) # local variable f1

lw $t0,20($sp) # arg n
bne $t0,$zero, over
add $v0,$zero,$t0
j exit

over: addi $s0, $zero,1
sw $s0,0($sp) # local variable i

lw $s1,20($sp) # arg n
```

```
lw $s3,8($sp) # f1 --> $s3
slt $t0, $s0,$s1
beq $t0,$zero, ileri
lw $s2,12($sp) # f0 --> $s2
```

```
geri: add $s4,$s2,$s3 # x --> $s4
add $s2,$s3,$zero
add $s3,$s4,$zero
addi $s0,$s0,1
slt $t0,$s0,$s1
bne $t0,$zero,geri
ileri: add $v0,$s3,$zero
```

```
exit: lw $ra, 16($sp)
      addi $sp,$sp,20
      jr $ra
```