```
#Chuong trinh: for-Fibonacci (pointer)
#----------------------------
          .include   "macro.mac"
#Data segment
          .data
#Cac dinh nghia bien
int_f:    .space     80   #20 phan tu word
int_n:    .word      13
#Cac cau nhac nhap du lieu
nhap_n:   .asciiz    "Nhap n: "
xuat_f1:  .asciiz    "Fibo("
xuat_f2:  .asciiz    ") = "
daungan:  .asciiz    "\n***********************\n"
#----------------------------
#Code segment
          .text
main:
#Nhap (syscall)
#Xu ly
  #f[0]=0
          la         $a1,int_f       #fptr=addr(f[0])
          sw         $zero,0($a1)
  #f[1]=1
          addi       $t0,$zero,1
          sw         $t0,4($a1)
  # a1=addr(f[i]),t0=f[i],t1=f[i-1],t2=f[i-2],t3=i,t4=20
  # for(i=2;i<20;i++)
          addi       $t3,$zero,2     #i=2
          addi       $a1,$a1,8       #fptr=addr(f[2])
          addi       $t4,$zero,20
  #fcond
fcond:    beq        $t3,$t4,endfor
  #fbody
    # f[i]=f[i-1]+f[i-2]
          lw         $t1,-4($a1)     #f[i-1]
          lw         $t2,-8($a1)     #f[i-2]
          add        $t0,$t1,$t2
          sw         $t0,0($a1)      #cat f[0]
  #floop
          addi       $t3,$t3,1       #i++
          addi       $a1,$a1,4       #fptr+=4
          j          fcond
  #endfor
endfor:
#Xuat ket qua (syscall)
lamlai:
  #Nhap n
          geti_p     nhap_n,int_n
  #Xuat ket qua
          puti_p     xuat_f1,int_n
          puts       xuat_f2
          lw         $t0,int_n       #n
          addi       $t1,$zero,4     #n*4
          mul        $t0,$t0,$t1
```

```
        la          $a1,int_f       #addr(f[0])
        add         $a1,$a1,$t0     #addr(f[n])
        lw          $a0,0($a1)
        addi        $v0,$zero,1
        syscall
# lam lai
        puts        daungan
        j           lamlai
#ket thuc chuong trinh (syscall)
Kthuc:  addi        $v0,$zero,10
        syscall
#-----------------------------
```