

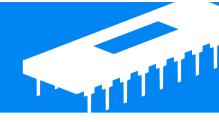
PER GLI STUDENTI



STUDIARE OGNI PROGRAMMA

E

ANALIZZARE L'EVOLUZIONE DELLO STACK (ANCHE CON L'USO DI DIGRAMMI E SCHEMI)



Si consideri la funzione f definita su interi

$$f(x) = f(x-2) - 2$$

$$f(1) = 14$$

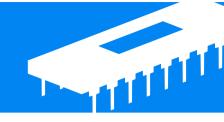
$$f(0) = 10$$

Si realizzi un programma in assembler MIPS che, definito un intero positivo $x \ge 2$, calcola il corrispondente valore di f(x) in modo **ricorsivo**

ESEMPIO:

$$f(6)=f(4)-2=(f(2)-2)-2=((f(0)-2)-2)-2=10-2-2-2=4$$

$$f(5)=f(3)-2=f(1)-2-2=14-2-2=10$$



Soluzione

.text

.globl main

main:

li \$v0,5 #lettura valore di ingresso

syscall

move \$a0,\$v0 #spostamento di X in registro preservante

jal Funzione #salto a funzione ricorsiva

move \$a0,\$v0 #recupero del valore di ritorno della funzione ricorsiva

li \$v0,1 #stampa del risultato

syscall

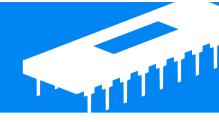
li \$v0, 10 #terminazione programma

syscall



Soluzione

```
Funzione:
          li $t1.2
                                         #confronto per stabilire se il valore analizzato fa riferimento al caso base
                                         #salto al caso base
          blt $a0, $t1, caso base
          subu $sp, $sp, 8
                                         #PUSH del valore X e dell'indirizzo di ritorno (nel primo caso al main; negli altri casi
                                         #a dopo la chiamata ricorsiva)
                                         #
          sw $a0, 0($sp)
          sw $ra, 4($sp)
          sub $a0, $a0, 2
                                         #aggiornamento del valore di X (x-2)
          ial Funzione
                                         #POP dei valori precedentemente custoditi nello stack
          lw $a0, 0($sp)
          lw $ra, 4($sp)
                                         #
          addi $sp, $sp, 8
          sub $v0,$v0,2
                                         #costante da sottrarre in base all'elemento selezionato
          ir $ra
                                         #salto per il POP successivo o per il ritorno del main
caso base:
          li $v0, 10
                                         #caso base X=0
          begz $a0,salta
                                         #caso base X=1
          li $v0,14
salta:
          jr $ra
                                         #salto per il POP o per il ritorno del main
```



Si consideri la funzione f definita su interi

$$f(x,y) = 2*f(x-2,y-5)$$

$$f(0,y) = 1$$

$$f(x,0) = 2$$

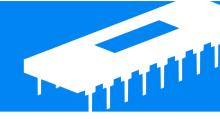
$$f(0,0)=3$$

Si realizzi un programma in assembler MIPS che, definiti due interi positivi $x \ge 2$ e $y \ge 2$, calcola il corrispondente valore di f(x,y) in modo **ricorsivo**

ESEMPIO:

$$f(4,25)=2*f(2,20)=2*2*f(0,15)=2*2*1=2*2*1=4$$

 $f(8,10)=2*f(6,5)=2*2*f(4,0)=2*2*2=8$
 $f(4,10)=2*f(2,5)=2*2*f(0,0)=2*2*3=12$



Soluzione

.text .globl main

main:

li \$v0,5 #lettura vprimo alore di ingresso syscall move \$a0,\$v0 #spostamento di X in registro preservante li \$v0,5 #lettura secondo valore di ingresso syscall move \$a1,\$v0 **#spostamento di Y in registro preservante** jal Funzione #salto a funzione ricorsiva move \$a0,\$v0 #recupero del valore di ritorno della funzione ricorsiva li \$v0,1 #stampa del risultato syscall

li \$v0, 10 syscall #terminazione programma

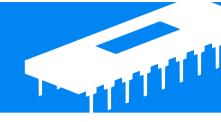


Soluzione

```
Funzione:
             begz $a0,caso base
                                        #confronto per stabilire il caso base
             begz $a1,caso base
                                       #PUSH del valore X e dell'indirizzo di ritorno (nel primo caso al main; negli altri casi a dopo la chiamata ricorsiva)
             subu $sp, $sp, 8
             sw $ra, 0($sp)
             sub $a0,$a0,2
                                       #aggiornamento X (x-2)
             sub $a1,$a1,5
                                       #aggiornamento Y (y-5)
             ial Funzione
                                       #POP dei valori di RA precedentemente custoditi nello stack
             lw $ra, 0($sp)
             addi $sp, $sp, 8
                                       #costante da moltiplicare all'elemento selezionato
             mul $v0,$v0,2
                                       #salto per il POP successivo o per il ritorno del main
             jr $ra
caso base:
             bnez $a0.zero2
             bnez $a1,zero1
             li $v0,3
             i finecasi
                                       \#x=0e\ y=0
zero1:
             li $v0,1
             i finecasi
zero2:
             li $v0,2
             i finecasi
                                       #non serve, ma lo scrivo per uniformità
```

finecasi:

jr \$ra



Si consideri la funzione f definita su interi

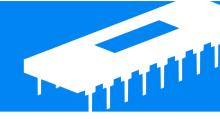
$$f(x,y) = (x-y)*f(x-5,y+3)$$

 $f(x,y) = 4 \text{ se } x \le 0$

Si realizzi un programma in assembler MIPS che, definiti due interi positivi e $x \ge 5$, calcola il corrispondente valore di f(x,y) in modo **ricorsivo**

ESEMPIO:

$$f(15,4)=11*f(10,7)=11*3*f(5,10)=11*3*-5*f(0,13)=11*3*-5*4=-660$$



Soluzione

.text .globl main

main:

li \$v0,5 #lettura vprimo alore di ingresso syscall move \$a0,\$v0 #spostamento di X in registro preservante li \$v0,5 #lettura secondo valore di ingresso syscall move \$a1,\$v0 **#spostamento di Y in registro preservante** jal Funzione #salto a funzione ricorsiva move \$a0,\$v0 #recupero del valore di ritorno della funzione ricorsiva li \$v0,1 #stampa del risultato syscall

li \$v0, 10 syscall #terminazione programma





Soluzione

```
Funzione:
```

```
blez $a0,casobase
          subi $sp,$sp,16
          sw $ra,0($sp)
          sw $a0,4($sp)
          sw $a1,8($sp)
                              \#x=x-5
          sub $a0,$a0,5
          add $a1,$a1,3
                              #y=y+3
          jal Funzione
          lw $ra,0($sp)
          lw $t0,4($sp)
          lw $t1,8($sp)
          add $sp,$sp,16
          sub $v1,$t0,$t1
                              #(x-y)
          mul $v0,$v0,$v1
                              #(x-y)*prodotti_precedenti
         jr $ra
casobase:
          li $v0,4
          jr $ra
```

