





Realizzare un programma in assembly MIPS che sposti tutti gli 1 a destra e gli 0 a sinistra della rappresentazione in binario di un operando intero.

ES:

INPUT: 358427306 (ovvero: 00010101011110100101010101010)

OUTPUT: 32767 (ovvero: 00000000000000011111111111111)

NB: L'operando intero a 32bit è immesso da tastiera. Il valore risultante deve essere riportato nel registro \$t0



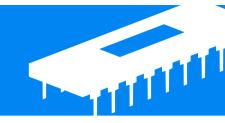
Realizzare un programma in assembly MIPS che risolva la funzione y così definita:

$$y = (\left\lfloor \sqrt{a} \right\rfloor + \left\lfloor \sqrt{b} \right\rfloor) \bmod (\left\lfloor \sqrt{a - b} \right\rfloor)$$

Gestire eventuali casi anomali

NB: La radice quadrata è la parte intera inferiore (es: $\sqrt{7}$ =2 e $\sqrt{9}$ =3 e $\sqrt{43}$ =6). I due valori - a, b - sono interi a 32bit letti da tastiera e la stampa del valore di y avviene su videoterminale





Realizzare un programma in assembly MIPS che immessi in input dei numeri interi calcola la media aritmetica (l'acquisizione dei dati termina con l'immissione dello zero). Supponendo che i valori immessi siano dei campioni relativi alla temperatura rilevata in un luogo e stabilito che la media nazionale è di 22.72°C calcolare l'errore quadratico metrico

$$E = \frac{\sqrt{|medianazionale^2 - mediarilevata^2|}}{2}$$

ESEMPIO:

INPUT:

22;23;25;27;22;23;21;18;0

OUTPUT

Media Rilevata: 22.625

E=1.037775





Matrice

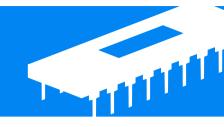
Data una matrice RxC realizzare un programma in asembly MIPS che restutisce la somma degli elementi aventi il valore uguale alla somma degli indici r e c

Esempio

$$a_{11}+a_{21}+a_{33}+a_{42}$$
 output=2+3+6+6=17

NB: La matrice è definita in memoria. Gli elementi della matrice sono interi a 16bit II programma deve valere per ogni matrice RxC, lo studente può inizializzare una generica matrice di dimensione fissa con R e C prefissati.





Funzione ricorsiva

Si consideri la funzione f definita su interi

$$f(x,y) = f(x-1,y-1) - 1$$
 se $x+y>0$
 $f(x,y) = 5$ se $x+y<=0$

Si realizzi un programma in assembler MIPS che, definiti due interi positivi x e y letti da tastiera (mediante sycall), calcoli il corrispondente valore di f(x,y) in modo ricorsivo utilizzando lo stack





Realizzare un programma in assembly MIPS che sposti tutti gli 1 a destra e gli 0 a sinistra della rappresentazione in binario di un operando intero.

ES:

INPUT: 358427306 (ovvero: 00010101011110100101010101010)

OUTPUT: 32767 (ovvero: 00000000000000011111111111111)

NB: L'operando intero a 32bit è immesso da tastiera. Il valore risultante deve essere riportato nel registro \$t0

SOLUZIONE TRACCIA 1

	.text	#Direttiva del Segmento Testo
main:	.globl main	#Direttiva per indicare l'etichetta main come globale
		#Etichetta main: inizio del programma
	li \$t0,0	#Inizializzazione registro con risultato
	li \$v0,5	#Lettura intero
	syscall	#
	move \$t2,\$v0	#Recupero dell'intero letto
ciclo:		
	andi \$t1,\$t2,1	#Estrazione bit meno significativo
	beqz \$t1, salta	#Analisi bit meno significativo
	sII \$t0,\$t0,1	#Shift a sinistra del contenuto di \$t0
	add \$t0,\$t0,1	#Settaggio ad 1 del bit meno significativo
salta:		
	srl \$t2,\$t2,1	#Shift dell'operando
	bnez \$t2,ciclo	#Ripetzione del ciclo fino a quando l'operando non è uguale a zero
	li \$v0,10 syscall	#Terminazione del programma



Realizzare un programma in assembly MIPS che risolva la funzione y così definita:

$$y = (\left\lfloor \sqrt{a} \right\rfloor + \left\lfloor \sqrt{b} \right\rfloor) \bmod (\left\lfloor \sqrt{a - b} \right\rfloor)$$

Gestire eventuali casi anomali

NB: La radice quadrata è la parte intera inferiore (es: $\sqrt{7}$ =2 e $\sqrt{9}$ =3 e $\sqrt{43}$ =6). I due valori - a, b - sono interi a 32bit letti da tastiera e la stampa del valore di y avviene su videoterminale

#Direttiva del Segmento Testo .text .globl main #Direttiva per indicare l'etichetta main come globale #Etichetta main: inizio del programma li \$v0.5 #Lettura A syscall move \$s0.\$v0 li \$v0.5 #Lettura B syscall move \$s1,\$v0 sub \$s2.\$s0.\$s1 #Calcolo A-B bltz \$s0, errore #Valutazione del caso in cui un argomento della radice sia negativa #Valutazione del caso in cui un argomento della radice sia negativa bltz \$s1, errore #Valutazione del caso in cui un argomento della radice sia negativa blez \$s2. errore #Calcolo SQRT INF(A) move \$a0,\$s0 ial RADICE QUADRATA move \$s0.\$v0 move \$a0,\$s1 #Calcolo SQRT_INF(B) ial RADICE QUADRATA move \$s1.\$v0

#Calcolo SQRT INF(A-B)

#Stampa del risultato

#Calcolo Z= SQRT INF(A)+SQRT INF(B)

#Calcolo Z=SQRT INF(A-B)

#Stampa stinga di errore

RADICE_QUADRATA:

li \$v0,0 #Registro in cui riportare il risultato

RQ_ciclo:

#Calcolo della parte intera inferiore della radice

#quadrata di un operando intero

mul \$t1,\$v0,\$v0

bgt \$t1,\$a0,RQ_fine

add \$v0,\$v0,1

j RQ_ciclo

RQ_fine:

SOLUZIONE TRACCIA 2

errore:

fine:

main:

la \$a0, msgTxt

move \$a0.\$s2

move \$s2,\$v0

add \$t0,\$s0,\$s1

rem \$a0,\$t0,\$s2

ial RADICE QUADRATA

li \$v0,4 syscall

li \$v0,1

syscall

i fine

li \$v0,10 #Terminazione del programma

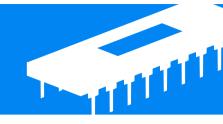
syscall

.data

msgTxt: .asciiz "Operando con valore negativo: la parte intera della radice quadrata non può essere calcolata o secondo termine del modulo nullo"

sub \$v0,\$v0,1

jr \$ra



Realizzare un programma in assembly MIPS che immessi in input dei numeri interi calcola la media aritmetica (l'acquisizione dei dati termina con l'immissione dello zero). Supponendo che i valori immessi siano dei campioni relativi alla temperatura rilevata in un luogo e stabilito che la media nazionale è di 22.72°C calcolare l'errore quadratico metrico e stamparlo su videoterminale

$$E = \frac{\sqrt{|medianazionale^2 - mediarilevata^2|}}{2}$$

ESEMPIO:

INPUT:

22;23;25;27;22;23;21;18;0

OUTPUT

(Media Rilevata: 22.625)

E=1.037775

.text **#Direttiva del Segmento Testo** .globl main #Direttiva per indicare l'etichetta main come globale #Etichetta main: inizio del programma li \$t0.0 li \$t1.-1 add \$t1.\$t1.1 #Contatore elementi immessi da tastiera #Lettura i-esimo elemento li \$v0,5 syscall add \$t0,\$t0,\$v0 #Totalizzatore per il calcolo della media bnez \$v0,ciclo mtc1 \$t0,\$f0 #Spostamento totalizzatore nel coprocessore mtc1 \$t1,\$f4 #Spostamento numero elementi campionati nel coprocessore cvt.s.w \$f0,\$f0 #Conversione valore #Conversione valore cvt.s.w \$f4,\$f4 div.s \$f0,\$f0,\$f4 #Media rilevata I.s \$f1.media nazionale #Lettura costanti I.s \$f2.due mul.s \$f0,\$f0,\$f0 #Media rilevata al quadrato mul.s \$f1,\$f1,\$f1 #Media nazionale al quadrato **#Determinante radice** sub.s \$f12,\$f1,\$f0 **#Valore** assoluto del dwteriminante abs.s \$f12,\$f12 sqrt.s \$f12,\$f12 #Radice del determinante div.s \$f12,\$f12,\$f2 #Errore quatratico li \$v0,2 **#Stampa** syscall li \$v0,10 #Terminazione del programma

main:

ciclo:

svscall

SOLUZIONE TRACCIA 3

.data

media nazionale:.float 22.72

due:.float 2.0





Matrice

Data una matrice RxC realizzare un programma in asembly MIPS che restutisce la somma degli elementi aventi il valore uguale alla somma degli indici r e c

Esempio

$$a_{11}+a_{21}+a_{33}+a_{42}$$
 output=2+3+6+6=17

NB: La matrice è definita in memoria. Gli elementi della matrice sono interi a 16bit II programma deve valere per ogni matrice RxC, lo studente può inizializzare una generica matrice di dimensione fissa con R e C prefissati.

```
#Direttiva per indicare l'etichetta main come globale
           .globl main
           li $t0,1 #indice i
           li $t1,1 #indice j
           lw $t2,R #numero righe
           lw $t3,C #numero colonne
analisi riga:
          li $t1.1
analisi colonna:
           sub $t6,$t0,1
                                 #Calcolo posizione elemento
          mul $t9,$t6,$t3
                                 #
                                 #
          sub $t7,$t1,1
           add $t9,$t9,$t7
          mul $t9,$t9,2
                                 #Prodotto per la dimensione degli elementi della matrice
           Ih $t8,matrice($t9)
           addi $t1,$t1,1
           ble $t1,$t3, analisi_colonna
           addi $t0,$t0,1
           ble $t0,$t2, analisi riga
          move $a0,$v0
          li $v0,1
           syscall
          li $v0,10
                                 #Terminazione programma
           syscall
```

#Direttiva del Segmento Testo

.text

main:

SOLUZIONE TRACCIA 4

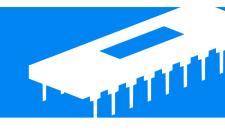
add \$t4, \$t0,\$t1 bne \$t8, \$t4, non_sommare add \$v0.\$v0.\$t8 **#Totalizzatore** non sommare:

.data

matrice: .word 2,5,7,3,0,8,7,9,6,1,6,5

R: .word 4 C: .word 3





Funzione ricorsiva

Si consideri la funzione f definita su interi

$$f(x,y) = f(x-1,y-1) - 1$$
 se $x+y>0$
 $f(x,y) = 5$ se $x+y<=0$

ESEMPIO
$$f(10,5)=f(9,4)-1=(f(8,3)-1)-1=((f(7,2)-1)-1)-1=(f(6,1)-1)-3=(f(5,0)-1)-4=(f(4,-1)-1)-5=(f(3,-2)-1)-6)=f(2,-3)-1)-7=5-8=-3$$

Realizzare un programma in assembler MIPS che, acquisiti due interi positivi x e y da tastiera (mediante sycall), calcola il corrispondente valore di f(x,y) in modo ricorsivo utilizzando lo stack e lo stampa su videoterminale

SOLUZIONE TRACCIA 5

allo studente

Commento in incarico

main:

.text

syscall

.globl main

li \$v0,5 **#Lettura X** syscall move \$a0,\$v0 #Lettura Y li \$v0,5 syscall move \$a1,\$v0 jal REC_FUNCTION #Salto a funzione ricorsiva move \$a0,\$v0 **#Stampa del risultato** li \$v0,1 syscall li \$v0,10 **#Terminazione programma**

#Direttiva del Segmento Testo

#Direttiva per indicare l'etichetta main come globale

REC_FUNCTION:

add \$t0.\$a0.\$a1 blez \$t0, caso base subu \$sp, \$sp, 8 sw \$ra, 4(\$sp) sub \$a0, \$a0, 1 sub \$a1, \$a1, 1 jal REC_FUNCTION sub \$v0, \$v0, 1 lw \$ra, 4(\$sp) addi \$sp, \$sp, 8 jr \$ra

caso base:

li \$v0, 5 jr \$ra

