Umut Akös
21202015
Pre-lab06/Section-2

Question-1

No.	Cache Size KB	N way cache	Word Size	Block size (no. of words)	No. of Sets	Tag Size in bits	Index Size (Set No.) in bits	Word Block Offset Size in bits[1]	Byte Offset Size in bits[2]	Block Replacement Policy Needed (Yes/No)
1	256	1	32 bits	4	2^{14}	14	14	2	2	NO
2	256	2	32 bits	4	2^{13}	15	13	2	2	YES
3	256	4	32 bits	8	2^{11}	16	11	3	2	YES
4	256	Full	32 bits	8	1	27	0	3	2	YES
9	512	1	16 bits	4	2^{16}	13	16	2	1	NO
10	512	2	16 bits	4	2^{15}	14	15	2	1	YES
11	512	4	16 bits	16	2^{12}	15	12	4	1	YES
12	512	Full	16 bits	16	1	27	0	4	1	YES

Question-2

Memory Address Accessed (hex)	Set No.	Hit (yes/no)
00 00 00 28	01	NO
00 00 00 49	01	NO
00 00 00 6C	01	NO
00 00 00 0C	01	NO
00 00 00 0B	01	YES
00 00 00 0D	01	YES

Question-3

Memory Address Accessed (hex)	Set No.	Hit (yes/no)
00 00 00 28	01	NO
00 00 00 49	01	NO
00 00 00 4C	01	YES
00 00 00 0C	01	NO
00 00 00 0B	01	YES
00 00 00 0D	01	YES

Question-4

Question-5

```
	.data
	prompt1:  .asciiz "Matrix size(N): "
	prompt2: .asciiz "Please enter the value to fill matrix: "
	prompt3:  .asciiz "Please enter the i value: "
	prompt4: .asciiz "Please enter the j value: "

	prompt5: .asciiz "1-Obtain summation of matrix elements row-major (row by row) summation,"
	prompt6: .asciiz "2-Obtain summation of matrix elements column-major (column by column) summation,"
	prompt7: .asciiz "3-Display desired elements of the matrix by specifying its row and column member"
	prompt8: .asciiz "4-exit"
	space2: .asciiz "\n"
	space3:		.asciiz " "


.text
	#give prompt to user to enter the Matrix size
	li $v0,4
	la $a0,prompt1
	syscall
	li $v0,5 #read integer for Matrix size
	syscall
	move $t0,$v0
	add $s3,$t0,$0
	mul $s0,$t0,$t0 #$s0 = matrix size

	#create array
	addi $v0,$0,9
	li $a0,36
	syscall
	addi $t1,$0,9   #keeps array size
	move $s6,$v0 #s6 contains adress of array
	addi $s7,$s6,0		#s7 to keep $s0


	addi $t2,$0,0 #counter

loop1:
	#give prompt to user to enter the number
	li $v0,4
	la $a0,prompt2
	syscall
	#get the user's number
	li $v0,5 #read integer
	syscall
	#store the number in $t0
	move $t0,$v0
	sw $t0, 0($s7)
	#check all array is full or not
	addi $s7,$s7,4
	addi $t2,$t2,1
	bne $t1,$t2,loop1
	# if array is full
	add $t2, $0, $0			# counter
	addi $s7,$s6, 0				#back to the array begin adress.

	#MENU
menu:
	#give prompt menu1
	li $v0,4
	la $a0,space2
	syscall
	li $v0,4
	la $a0,prompt5
	syscall
	li $v0,4
	la $a0,space2
	syscall
	#give prompt to menu2
	li $v0,4
	la $a0,prompt6
	syscall
	li $v0,4
	la $a0,space2
	syscall
	#give prompt to user to menu3
	li $v0,4
	la $a0,prompt7
	syscall
	li $v0,4
	la $a0,space2
	syscall
	#give prompt to user to exit
	li $v0,4
	la $a0,prompt8
	syscall
	li $v0,4
	la $a0,space2
	syscall
	#get the user's number
	li $v0,5 #read integer
	syscall

	addi $t0,$0,1
	addi $t1,$0,2
	addi $t2,$0,3
	addi $t3,$0,4
	#store the number in $s7
	move $s5,$v0
	beq $s5,$t0,obtainSumRowRow2
	beq $s5,$t1,obtainSumColCol2
	beq $s5,$t2,DisplayEl
	beq $s5,$t3,exit

DisplayEl:
	#give prompt2 and prompt3 to user to enter the i,j
	li $v0,4
	la $a0,prompt3
	syscall
	li $v0,5 #read integer for i value
	syscall
	move $s1,$v0  #$s1 = i

	li $v0,4
	la $a0,prompt4
	syscall
	li $v0,5 #read integer for j value
	syscall
	move $s2,$v0 #$s2 = j

	#FORMULA (i - 1) x N x 4 + (j - 1) x 4 ### s3 = N ,s1 = i , s2 = j
	addi $t0,$s1,-1
	mul $t0,$t0,4
	mul $t0,$t0,$s3
	addi $t1,$s2,-1
	mul $t1,$t1,4
	add $t3,$t1,$t0
smallLoop1:
	ble $t3,0,printij
	addi $s7,$s7,4
	addi $t3,$t3,-4
	j smallLoop1
printij:
	lw $t0,0($s7)
	li $v0,1
	move $a0,$t0
	syscall
	addi $s7,$s6,0
```

```
    j menu

#all row start as 1's but s2 will be later
obtainSumRowRow2:
    addi $s1,$0,1
    addi $s2,$0,0

obtainSumRowRow:
    addi $s2,$s2,1
    ble $s2,$s3,findRowElement
    j printSumRowRow
continue1:
    addi $s1,$s1,1
    addi $s2,$0,0
    add $s4,$0,$0
    ble  $s1,$s3,jumway
    j menu
jumway:            #jumway provides new row elements
    j obtainSumRowRow
findRowElement:
    addi $t0,$s1,-1
    mul $t0,$t0,4
    mul $t0,$t0,$s3
    addi $t1,$s2,-1
    mul $t1,$t1,4
    add $t3,$t1,$t0
smallLoop2:
    ble $t3,0,sumRow
    addi $s7,$s7,4
    addi $t3,$t3,-4
    j smallLoop2
sumRow:
    lw $t0,0($s7)
    add $s4,$s4,$t0
    add $s7,$s6,$0
    j obtainSumRowRow
printSumRowRow:                    #to take row sum
    li $v0,1
    move $a0,$s4
    syscall
    li $v0,4
    la $a0,space2
    syscall
    j continue1
##########################################Column
sum#################################3
#all row start as 1's but s2 will be later
obtainSumColCol2:
    addi $s2,$0,1
```

```
	addi $s1,$0,0

obtainSumColCol:
	addi $s1,$s1,1
	ble $s1,$s3,findColElement
	j printSumColCol
continue1C:
	addi $s2,$s2,1
	addi $s1,$0,0
	add $s4,$0,$0
	ble  $s2,$s3,jumwayC
	j menu
jumwayC:		#jumway provides new col elements
	j obtainSumColCol
findColElement:
	addi $t0,$s1,-1
	mul $t0,$t0,4
	mul $t0,$t0,$s3
	addi $t1,$s2,-1
	mul $t1,$t1,4
	add $t3,$t1,$t0
smallLoop2C:
	ble $t3,0,sumCol
	addi $s7,$s7,4
	addi $t3,$t3,-4
	j smallLoop2C
sumCol:
	lw $t0,0($s7)
	add $s4,$s4,$t0
	add $s7,$s6,$0
	j obtainSumColCol
printSumColCol:				#to take col sum
	li $v0,1
	move $a0,$s4
	syscall
	li $v0,4
	la $a0,space2
	syscall
	j continue1C
exit:
	li $v0,10
	syscall
```