Winston Shih
WXS190012
CS 2340.003

CS 2340 Assignment 3

Q1.

Edit | Execute

CS2340HW3Question1.asm

```
#Winston Shih
#WXS190012
#CS 2340.003
.data #Section represents data section of machine code.
     x: .word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 #Stores the x array.
     message: .asciiz "Element at x[" #Stores "Element at x[" string.
     arraybrackets: .asciiz "][" #Stores "][" string.
     arraybracket: .asciiz "] = " #Stores "] = " string.
     newline: .asciiz "\n" #Stores new line.
.text #Section makes machine code executable.
main: #Represents the main(void) method of program.
     addi $s0, $zero, 2 #Sets total number of rows equal to 2
     addi $s1, $zero, 3 #Sets total number of columns equal to 3.
     addi $s2, $zero, 2 #Sets total depth to 2.
     addi $t0, $zero, 0 #Initializes i to 0.
     outerloop: slti $s3, $t0, 2 #Checks to see if i<2 and stores result in $s3.
                beq $s3, $zero, exit #If $s3=0, then i<2 is false and program jumps to exit.
                addi $t1, $zero, 0 #int j=0
     middleloop: slti $s4, $t1, 3 #Checks to see if j<3 and stores result in $s4.
                 beq $s4, $zero, exit2 #If $s4=0, then j<3 is false and jump to exit2.
                 addi $t2, $zero, 0 #Sets k=0.
    innerloop: slti $s5, $t2, 2 #Checks to see if k<2 and stores result in $s5.
               beq $s5, $zero, exit3 #If $s5=0, then condition k<2 is false and jump to exit3.
               li $v0, 4 #Requests print string service.
               la $a0, message #Loads message address to $a0.
               syscall #Prints "Element at x[".
               li $v0, 1 #Requests print string service
               move $a0, $t0 #Moves index i's address to $a0.
               syscall #Prints index i.
               li $v0, 4 #Requests print string service.
               la $a0, arraybrackets #Loads address of arraybrackets to $a0.
               syscall #Prints "]["
               li $v0, 1 #Requests print string service.
               move $a0, $t1 #Loads address for index j.
               syscall #Prints index j.
```

Edit | Execute

CS2340HW3Question1.asm

```
               syscall #Prints index j.
               li $v0, 4 #Requests print string service.
               la $a0, arraybrackets #Loads address of arraybrackets to $a0.
               syscall #Prints "]["
               li $v0, 1 #Requests print string service.
               move $a0, $t2 #prints index k.
               syscall #Prints index k.
               li $v0, 4 #Requests service to print string.
               la $a0, arraybracket #Loads address of arraybracket to $a0.
               syscall #Prints "] = "
               mul $t3, $t0, $s1 #$t3=i*total columns
               mul $t3, $t3, $s2 #$t3=i*total columns*total depth
               mul $t4, $t1, $s2 #$t4=j*total depth
               add $t5, $t3, $t4 #$t5=i*total columns*total depth+j*total depth
               add $t5, $t5, $t2 #$t5=i*total columns*total depth+j*total depth+k
               li $v0, 1 #Requests print integer service.
               move $a0, $t5 #Moves address of x[i][j][k] to $a0.
               syscall #Prints number located at x[i][j][k].
               li $v0, 4  #Requests print string service.
               la $a0, newline #Loads address of newline ro $a0.
               syscall #Prints newline.
               add $t2, $t2, 1 #Increments k by 1.
               j innerloop #Jumps back to beginning of innerloop.
    exit3: addi $t1, $t1, 1 #Increments j by 1.
           j middleloop #Jumps back to beginning of middleloop.
    exit2: addi $t0, $t0, 1 #Increments i by 1.
           j outerloop #Jumps back to beginning of outerloop.
    exit: li $v0, 10 #Requests service to end program.
          syscall #Terminates the program.
```

Mars Messages | Run I/O

Clear

```
Element at x[0][0][0] = 0
Element at x[0][0][1] = 1
Element at x[0][1][0] = 2
Element at x[0][1][1] = 3
Element at x[0][2][0] = 4
Element at x[0][2][1] = 5
Element at x[1][0][0] = 6
Element at x[1][0][1] = 7
Element at x[1][1][0] = 8
```

Mars Messages | Run I/O

Clear

```
Element at x[1][0][1] = 7
Element at x[1][1][0] = 8
Element at x[1][1][1] = 9
Element at x[1][2][0] = 10
Element at x[1][2][1] = 11

-- program is finished running --
```

Q2.

Edit | Execute

CS2340HW3Question2.asm

```
#Winston Shih
#WXS190012
#CS 2340.003
.data #This represents data section of the program.
     s: .asciiz "grrksfoegrrks" #Array s stores "grrksfoegrrks".
     c1: .byte 'e' #c1 stores char 'e'.
     c2: .byte 'r' #c2 stores char 'r'.
.text #This keyword makes the program executable.
     main: #Represents the main function of program.
          la $a0, s #Loads address of "grrksfoegrrks" to $a0.
          lb $a1, c1 #Loads byte of 'e' to $a1.
          lb $a2, c2 #Loads byte of 'r' to
          jal replace #Calls replace function.
          li $v0, 4 #Loads print string service.
          la $a0, s #Loads address of
          syscall #Prints the new string stored in s array.
          li $v0, 10 #Requests service to terminate program.
          syscall #Ends the program.
    replace: #Represents the replace method.
            subi $sp, $sp, 8 #Adjusts stack pointer to have space to push $ra into stack.
            sw $ra, ($sp) #Pushes $ra in stack and stores its address.
            la $a3, s #Loads address of "grrksfoegrrks" to $a3.
            addi $t0, $zero, -1 #Sets index for strlen loop to -1, so loop runs at least once.
    strlen: lb $t1, ($a3) #Loads "grrksfoegrrks" from $a3 to $t1.
            addi $a3, $a3, 1 #Increments string pointer by 1.
            addi $t0, $t0, 1 #Increments i by 1.
            bne $t1, $zero, strlen #If s[i] is not null, then program loops back to beginning of strlen.
    addi $t2, $zero, 0 #Initializes i to 0.
    forloop: lb $t3, 0($a0) #Loads "grrksfoegrrks"'s byte into $t3.
             beq $t2, $t0, complete  #If i=1, then jump to complete.
             beq $t3, $a1, replacetor #If s[i]==c1, then jump to replacetor.
             beq $t3, $a2, replacetoe #If s[i]==c2, then jump to replacetoe
             addi $a0, $a0, 1 #Increments string pointer by 1.
             addi $t2, $t2, 1 #Increments i by 1.
             j forloop #Jumps to beginning of forloop.
```

```
36      replacetor: sb $a2, 0($a0) #Sets s[i] equal to c2.
37                  addi $a0, $a0, 1 #Increments string pointer by 1.
38                  addi $t2, $t2, 1 #Increments i by 1.
39                  j forloop #Jumps to forloop.
40      replacetoe: sb $a1, 0($a0) #Sets s[i] equal to c1.
41                  addi $a0, $a0, 1 #Increments string pointer by 1.
42                  addi $t2, $t2, 1 #Increments i by 1.
43                  j forloop #Jumps to forloop.
44      complete: lw $ra, ($sp) #Restores return address of replace function.
45                addi $sp, $sp, 8 #Pops replace function's return address out of stack.
46                jr $ra #Returns result of replace method.
```

Line: 46 Column: 55 ☑ Show Line Numbers

Mars Messages | Run I/O

```
geeksforgeeks
-- program is finished running --
```

Q3.

Edit | Execute

CS2340HW3Question3.asm

```
1   #Winston Shih
2   #WXS190012
3   #CS 2340.003
4   .data #Represents the data section of machine code.
5        arr: .space 64 #4*4 matrix has 16 integers*4 bytes/integer= 64 total bytes.
6   .text #Key word that makes machine code executable.
7        addi $s0, $zero, 4 #M=4.
8        addi $s1, $zero, 4 #N=4.
9        main: #Represents the main methof of program.
10            la $a0, arr #Loads arr's address to $a0.
11            addi $t0, $zero, 1 #Initializes x to 1.
12            addi $t1, $zero, 0 #Initializes i to 0.
13       mainouterloop: beq $t1, $s0, printsum #Checks to see if i equals M. If true,program jumps to printsum.
14                      addi $t2, $zero, 0 #Initializes j to 0.
15       maininnerloop: beq $t2, $s1, nextimain #Jumps to nextimain if j=N.
16                      mul $t3, $t1, $s1 #$t3=i*N
17                      add $t3, $t3, $t2 #$t3=i*N+j
18                      sll $t3, $t3, 2 #Creates offset for arr[i][j].
19                      add $t3, $t3, $a0 #Creates arr[i][j]'s address.
20                      sw $t0, ($t3) #Stores x's value to arr[i][j].
21                      addi $t0, $t0, 1 #x is incremented by 1.
22                      addi $t2, $t2, 1 #j is incremented by 1.
23                      j maininnerloop #Jumps back to beginning of maininnerloop.
24       nextimain: addi $t1, $t1, 1 #i is incremented by 1.
25                  j mainouterloop #Jumps to mainouterloop.
26       printsum: la $a0, arr #Loads address of arr array to $a0.
27                 jal sum #Calls back sum function.
28                 move $a0, $v0 #Moves return value of sum from $v0 to $a0.
29                 li $v0, 1 #Requests print integer service.
30                 syscall #Prints value of sum.
31                 li $v0, 10 #Requests service to end program.
32                 syscall #Ends program.
33       sum: addi $t4, $zero, 0 #Initializes sum to 0.
34            addi $t1, $zero, 0 #Initializes i to 0.
35       sumouterloop: beq $t1, $s0, exit #Jumps to exit.
```

```
36                  sub $t2, $t2, $t2 #initializes j to 0.
37      suminnerloop: beq $t2, $s1, nextisum #Jumps to nextisum.
38                  mul $t5, $t1, $s1 #$t5=i*N
39                  add $t5, $t5, $t2 #$t5=i*N+j
40                  sll $t5, $t5, 2 #Creates offset for arr[i][j].
41                  add $t5, $t5, $a0 #Creates address for arr[i][j]
42                  lw $t6, ($t5) #Loads arr[i][j]'s address to $t6.
43                  add $t4, $t4, $t6 #sum=sum+arr[i][j]
44                  addi $t2, $t2, 1 #j increments by 1.
45                  j suminnerloop #Jumps to suminnerloop.
46      nextisum: addi $t1, $t1, 1 #i increments by 1.
47              j sumouterloop #Jumps to sumouterloop.
48      exit: move $v0, $t4 #Moves sum to register $v0.
49            jr $ra #Returns the sum.
```

Line: 49 Column: 38 ☑ Show Line Numbers

Mars Messages | Run I/O

```
136
-- program is finished running --
```