

CS 224

Section No. 4

Spring 2020

Lab No. 1

Mannan Abdul

21801066

Array Program

Mannan Abdul Lab01 Part01a

.text

readSize:

la \$a0, sizeMsg #Output prompt to get the size from the user

li \$v0, 4

syscall

li \$v0, 5 #We read the user specified size of the array

syscall

bgt \$v0, 20, readSize #Asks the user to enter the array size again if it is larger than 20

sw \$v0, size

la \$t0, array #Loads the address of array to \$t0

lw \$t1, size #Loads size to \$t1

li \$t2, 0 #\$t2 will be used to compute the sum of all the integers in the array

readArray:

la \$a0, msg #Output prompt to get the array element from the user

li \$v0, 4

syscall

li \$v0, 5 #We read the user specified element of the array

syscall

sw \$v0, (\$t0) #The array element is saved to the first memory address reserved for the array

add \$t2, \$t2, \$v0 #the sum of the integers is incremented

addi \$t0, \$t0, 4 #Adds 4 to \$t0 to update the address to save the next element

addi \$t1, \$t1, -1 #Keeps count so we only make an array of the size specified

bgt \$t1, \$zero, readArray #Checks if all values have been taken, else goes back to readArray

la \$a0, arrayMsg #Displays array contents one by one after msg

li \$v0, 4

syscall

la \$t0, array #Loads array address to start displaying them

lw \$t1, size #Loads size again to help keep count

print:

lw \$t3, 0(\$t0) #Loads elements to \$t3 to display to the user

la \$a0, endl #Goes to new line after every element displayed

li \$v0, 4

syscall

li \$v0, 1 #Array element is displayed

add \$a0, \$zero, \$t3

syscall

addi \$t0, \$t0, 4 #Adds 4 to \$t0 to update the address to display the next element

```
addi $t1, $t1, -1 #Keeps count to help display the array
```

```
bgt $t1, $zero, print #Checks if all elements are displayed, else will keep printing
```

```
la $a0, endl #Goes to new line before displaying the sum
```

```
li $v0, 4
```

```
syscall
```

```
la $a0, sum #Gives msg that it will display the sum
```

```
li $v0, 4
```

```
syscall
```

```
li $v0, 1 #Displays the sum
```

```
add $a0, $zero, $t2
```

```
syscall
```

```
li $v0, 10 #Program is done
```

```
syscall
```

```
.data
```

```
array:      .space 80
```

```
size:       .space 4
```

```
sizeMsg:    .asciiz "\n Enter the size of the array (size <= 20): "
```

```
msg:        .asciiz "\n Enter the Array Element: "
```

```
arrayMsg:   .asciiz "\n Array Contents: "
```

```
endl:       .asciiz "\n "
```

```
sum:        .asciiz "The sum of the contents is: "
```

Arithmetic Program

Mannan Abdul Lab01 Part01b

.text

li \$v0, 4 #Show user the expression and ask input for a

la \$a0, prompt

syscall

li \$v0, 5 #Take input for a

syscall

sw \$v0, a #Store the value of a

li \$v0, 4 #Ask input for b

la \$a0, bMsg

syscall

li \$v0, 5 #Take input for b

syscall

sw \$v0, b #Store input for b

li \$v0, 4 #Ask input for c

la \$a0, cMsg

syscall

li \$v0, 5 #Take input for c

syscall

sw \$v0, c #Store input for c

li \$v0, 4 #Ask input for d

la \$a0, dMsg

syscall

li \$v0, 5 #Take input for d

syscall

sw \$v0, d #Store input for d

#Load all values into argument registers

lw \$a0, a

lw \$a1, b

lw \$a2, c

lw \$a3, d

jal calculateExpression #Go to method to calculate ans

sw \$v0, x #Store the ans in x

li \$v0, 4 #Show msg for output of ans

la \$a0, msg

syscall

lw \$t0, x #Load ans into \$t0

li \$v0, 1 #Display the ans

add \$a0, \$zero, \$t0

syscall

li \$v0, 10 #Program is done

syscall

calculateExpression:

sub \$t0, \$a1, \$a2 # $t0 = b - c$

mul \$t1, \$t0, \$a0 # $t1 = a * (b - c)$

div \$t2, \$t1, \$a3 # $t2 = a * (b - c) / d$

mfhi \$t3 # $t3 = a * (b - c) \% d$

add \$v0, \$zero, \$t3 #Put value into \$v0 to return

jr \$ra #Go back to main method

.data

a: .space 4

b: .space 4

c: .space 4

d: .space 4

x: .space 4

prompt: .asciiz "For the Expression $x = a * (b - c) \% d$ \n Enter a: "

bMsg: .asciiz "\n Enter b: "

cMsg: .asciiz "\n Enter c: "

dMsg: .asciiz "\n Enter d: "

msg: .asciiz "\n The answer is: "