.data

array: .space 80

newLine: .asciiz "\n"

Spacing: .asciiz " "

perLine: .asciiz "\nHow many integers per line?: "

Prompt: .asciiz "\n Enter an integer: "

.globl main

.text

main: li $t0, 20 #$t0 keeps track of integers to be read.

la $t1, array #loads starting address of array

la $a0, perLine

li $v0, 4

syscall

li $v0, 5

syscall

add $t3, $t3, $v0 #variable holding integers per line

loopR: la $a0, Prompt

li $v0, 4 #System call code for Print String

syscall

li $v0, 5 #Reads an integer

syscall

sw $v0, 0($t1) #storing the integer entered

add $t0, $t0, -1 #Decrement the number of integers

add $t1, $t1, 4 #load the address of next integer

bgtz $t0, loopR #branch to read and store next int

li $t0, 20 #re-initializing the array

la $t1, array

loopP: lw $a0, ($t1) #load an integer from memory location to $a0

li $v0, 1

syscall #prints the integer

add $t0, $t0, -1

add $t1, $t1, 4 #Next array index

la $a0, newLine

li $v0, 4

syscall

bgtz $t0, loopP #continue loop until $t0 <= 0

li $t0, 20 #reinitializing array

la $t1, array

la $a0, newLine

li $v0, 4

syscall

loopQ: lw $a0, ($t1) #load an integer from memory location to $a0

li $v0, 1

syscall #prints the integer

add $t0, $t0, -1

add $t1, $t1, 4

la $a0, Spacing #Spacing out numbers

li $v0, 4

syscall

bgtz $t0, loopQ #Loop until $t0 <= 0

li $t0, 20 #re-initializing array

la $t1, array

la $a0, newLine

li $v0, 4

syscall

syscall

loopT: blez $t0, End #This marks the eventual end of the loop

li $t2, 0 #Makes sure the counter $t2 = 0

add $t2, $t2, $t3 #Adds integer per line to counter

blez $t3, End #Incase integer per line = 0

jal loopS #IFF all conditions are satisfied, go to inner loop

loopS: blez $t0, End #No more elements, finish loop

lw $a0, ($t1)

li $v0, 1

syscall

la $a0, Spacing

li $v0, 4

syscall

addi $t2, $t2, -1 #Decrementing per line counter

addi $t1, $t1, 4 #incrementing array index

addi $t0, $t0, -1 #Decrementing element counter

bgtz $t2, loopS #Keep looping while $t2 > 0

la $a0, newLine

li $v0, 4

syscall

blez $t2, loopT #Back to outer loop when $t2 <= 0

End: li $v0, 10 #terminate

syscall

How many integers per line?: 6

Enter an integer: 1

Enter an integer: 2

Enter an integer: 3

Enter an integer: 4

Enter an integer: 5

Enter an integer: 6

Enter an integer: 7

Enter an integer: 8

Enter an integer: 9

Enter an integer: 10

Enter an integer: 11

Enter an integer: 12

Enter an integer: 13

Enter an integer: 14

Enter an integer: 15

Enter an integer: 16

Enter an integer: 17

Enter an integer: 18

Enter an integer: 19

Enter an integer: 20

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4 5 6

7 8 9 10 11 12

13 14 15 16 17 18

19 20

How many integers per line?: 2

Enter an integer: 563

Enter an integer: 77

Enter an integer: 553

Enter an integer: 676

Enter an integer: 3

Enter an integer: 6

Enter an integer: 3577

Enter an integer: 356

Enter an integer: 76

Enter an integer: 35

Enter an integer: 6735

Enter an integer: 67

Enter an integer: 3567

Enter an integer: 3567

Enter an integer: 57

Enter an integer: 653

Enter an integer: 756

Enter an integer: 7

Enter an integer: 3567

Enter an integer: 5

563

77

553

676

3

6

3577

356

76

35

6735

67

3567

3567

57

653

756

7

3567

5

563 77 553 676 3 6 3577 356 76 35 6735 67 3567 3567 57 653 756 7 3567 5

563 77

553 676

3 6

3577 356

76 35

6735 67

3567 3567

57 653

756 7

3567 5