Practical-10

Aim Using SPIM, write and test a program that reads in a positive integer using the SPIM system calls. If the integer is not positive, the program should terminate with the message "Invalid Entry"; otherwise the program should print out the names of the digits of the integers, delimited by exactly one space. For example, if the user entered "528," the output would be "Five Two Eight."

```
.data
str:
    .asciiz "Please input some number."
str0:
    .asciiz "Zero "
str1:
    .asciiz "One "
str2:
    .asciiz "Two "
str3:
    .asciiz "three "
str4:
    .asciiz "four
```

```
"
str5:
    .asciiz "Five "
str6:
    .asciiz "Six "
str7:
    .asciiz "Seven "
str8:
    .asciiz "Eight "
str9:
    .asciiz "Nine "
    .text
    .globl main
main:
li $v0, 4 # prints
la $a0, str# the
syscall# string
li $v0, 5 # read_int= 5
syscall# input
addi $s0, $v0, 0
li $t0, 0
li $t1, 1
loop1: # using thisloop we get the number of
digits
addi $t0, $t0, 1
```

```
beqz $v0, done1
li$t5, 10
mulou $t1, $t1, $t5
divu $v0, $v0, 10
j loop1
done1:
divu $t1, $t1, 10
li$t2, 0
loop2:
divu $t3, $s0, $t1 # get the next most significant
digit
li$t4, 0
beq $t3, $t4, printzero
addi $t4, $t4, 1
beq $t3, $t4, printone
addi $t4, $t4, 1
beq $t3, $t4, printtwo
addi $t4, $t4, 1
beq $t3, $t4, printthree
addi $t4, $t4, 1
beq $t3, $t4, printfour
addi $t4, $t4, 1
beq $t3, $t4, printfive
addi $t4, $t4, 1
beq $t3, $t4, printsix
```

```
addi $t4, $t4, 1
beq $t3, $t4, printseven
addi $t4, $t4, 1
beq $t3, $t4, printeight
addi $t4, $t4, 1
li $v0, 4 # prints
la $a0, str9 # the
syscall # string
j doneprint
printzero:
li $v0, 4 # prints
la $a0, str0 # the
syscall # string
j doneprint
printone:
li $v0, 4 # prints
la $a0, str1 # the
syscall # string
j doneprint
printtwo:
li $v0, 4 # prints
la $a0, str2 # the
syscall # string
j doneprint
printthree:
```

```
li$v0, 4 # prints
la$a0, str3# the
syscall# string
j doneprint
printfour:
li$v0, 4 # prints
la$a0, str4# the
syscall# string
j doneprint
printfive:
li$v0, 4 # prints
la$a0, str5# the
syscall# string
j doneprint
printsix:
li$v0, 4 # prints
la$a0, str6# the
syscall# string
j doneprint
printseven:
li$v0, 4 # prints
la$a0, str7# the
syscall# string
j doneprint
printeight:
```

```
li$v0, 4 # prints
la$a0, str8# the
syscall# string
doneprint:
addi $t2, $t2, 1
mulou $t3, $t3, $t1
subu $s0, $s0, $t3 # remove the most significant
digit
divu $t1, $t1, 10
bne $t0, $t2, loop2
```

Output :

```
Please input some number. 658
Six Five Eight
```