CSE 331 HW1 MIPS Assembly

My MIPS code firstly open the text file and stores the file pointer. Then reads the file and stores the length of the text. After that, holds a temporary counter and starts the loop.

loop:

Reads each byte of the text and jumps to check if the character is a number.

check:

Checks if the character is a number by calling isNum. If it is a number, jumps to isDigit to check if it is a single digit in the text stream. Unless, jumps to notNum.

isDigit:

Check if the number is a digit at the start of the text, at the end of the text, at the start of a sentence, at the end of a sentence or somewhere between. If it is a single digit, jumps to smallNum or bigNum to print the number corresponding to where it is placed. Unless, jumps to notNum.

smallNum/bigNum:

Checks which digit it is and prints it small/big with characters.

num:

Just increases the value of the counter since the character is already printed as a digit.

notNum:

Prints the character, increases the counter and iterates the loop.

smallZero, bigZero, …:

Just prints small zero (zero), big zero (Zero), etc. as a text.

exit:

Closes the file and exits the program.

Pseudocode:

file opens

file pointer is stored

file reads

text length is stored

loop:

condition for that if all the text checked; if all checked, then jumps to exit

reads the character and jumps to check

notNum:

prints the character

num:

iteration

jumps to loop

check:

checks if number (call the function isNum)

if number checks if digit (jumps isDigit)

jumps to notNum if it is not a number

isDigit:

check if it is a single digit, and if so, checks where it is in the text

jumps to smallNum/bigNum if it is a small/big digit depending on where it is and prints the num

jumps to notNum if it is not a number

smallNum/bigNum:

checks which digit it is and prints it small/big with characters

smallZero, bigZero, …:

prints small zero (zero), big zero (Zero), etc. as a text

jumps to num

exit:

closes the file

exits the program