CSE 331 Computer Organization
Project 1 – MIPS Assembly

151044076 – SEVGİ BAYANSALDUZ

First I wrote the c code of the problem.Then I translated the c code to mips assembly code.The C code is below:

```c
#include <stdio.h>
#include <string.h>

void main(){

    char *digits[]={"zero", "one","two","three","four","five","six","seven","eight","nine"};
    char *str=" 9I am 77 years old 2.0";
    int i,j,size2,k=0;
    int size=strlen(str);
    char newm[]="";
    for(i =0;i<size;i++)
    {
        if((str[i]<='9' && str[i]>='0' ) &&
         (i<=1 || str[i-2]>'9' || str[i-2]<'0' || str[i-1]!='.') &&
        (i==0 || str[i-1]>'9'  ||  str[i-1]<'0') &&
         (size-3<i || str[i+2]>'9' || str[i+2]<'0' || str[i+1]!='.')&&
        (i==size-1 || str[i+1]>'9'  || str[i+1]<'0')   )
        {
            size2=strlen(digits[str[i]-'0']);

            for(j=0;j<size2;j++,k++)
            {
                if((i==0|| (i==1 && str[i-1]==' ') || (str[i-1]==' ' && str[i-2]=='.')) && j==0)
                    newm[k]=digits[str[i]-'0'][j]-32;
                else newm[k]=digits[str[i]-'0'][j];
            }
        }else {
            newm[k]=str[i];
            k++;
        }
    }
    newm[k]='\0';
    printf("\n%s",newm);
}
```

The explanation of the c code above:

1. i = 0 to size of the input text. (loop cond.)
2. İf:
    - (purple arrow) If str[i] is a number (str is input text)
    - (light blue arrow) if i less or equal to 1 ,or str[i-2] is not a number or, str[i-1] is not a dot.
    - (green arrow) if i equals zero or str[i-1] is not a number
    - (grey arrow) if i is less or equal to size -2, or str[i+2] is not a number or, str[i+1] is not a dot.
    - (orange arrow) if i equals size-1 or str[i-1] is not a number

   If all conditions are met changes number to text instead of digits and go to if statement .Otherwise go to else.

   ➔j=0 to size of the digits.(Array digits includes representation of the digits)
   ➔if i equals zero (it means number is at the beginning of the paragraph) ,or i equals one and str[i-1] is space character,or str[i-1] is space character and str[i-2] is dot character.And j equals zero (it means does j show the first letter of the number). Make first letter of the text capital letter and store the text representation of the number into the string neww. Else store the text representation of digit without capital letter.
3. Else : Do not change the character and store this character into the string neww.
4. Add null character into the end of the string neww.

MIPS EXPLANATION

```
.data
fileStr: .space 224
newline: .byte '\n'
textData: .space 256
outData: .space 256
zero: .asciiz "zero"
one: .asciiz "one"
two: .asciiz "two"
three: .asciiz "three"
four: .asciiz "four"
five: .asciiz "five"
six: .asciiz "six"
seven:.asciiz "seven"
eight:.asciiz "eight"
nine:.asciiz "nine"
list: .word zero,one,two,three,four,five,six,seven,eight,nine
.text
.globl main
```

I save the text representations of the digits into the data list.

fileStr stores a file name which is taken as an input from a user.

textData stores the content of the file.

outData stores the modified content.

```
#####for loop
for1st: slt $t0,$s1,$s5 # if i<size $t0=1  (s5 saves the size of the textData)
        beq $t0,$zero,exit1#exit the for loop
####if
# Conditions in first parenthesis: (textData[i]<='9' && textData[i]>='0')
        add $t2,$s4,$s1 # $t2 = textData + i
        lb $t3 ,0($t2)  # $t3 = textData[i]
        slti $t0,$t3,58 # if textData[i]<='9' $t0=1
        beq $t0,$zero,else# go to
        slti $t0,$t3,48 # if textData[i]< 0 $t0=1
        bne $t0,$zero,else# go to else
#Conditions in second parenthesis :(i<=1 || textData[i-2]>'9' || textData[i-2]<'0' || textData[i-1]!='.')
        slti,$t0,$s1,2 # if i<2 t0=1
        bne $t0,$zero,cond3 # if a condition is met, switch to other parenthesis
        lb $t4 ,-2($t2)#t4=textData[i-2]
        slti $t0,$t4,58 #if textData[i-2]>'9' t0=0
        beq $t0,$zero,cond3 # if a condition is met, switch to other parenthesis
        slti $t0,$t4,48 # if textData[i-2]<'0' t0=1
        bne $t0,$zero,cond3 # if a condition is met, switch to other parenthesis
        lb $t4,-1($t2) # t4=textData[i-1]
        addi $t0,$zero,46# t0='.'
```

Label For1st include if and else labels. All the steps in the hw1.mips file are clearly specified.

PROCODURES

```
write:
  # Open (for writing) a file that does not exist
  li   $v0, 13       # system call for open file
  la   $a0, fileStr     # output file name
  li   $a1, 1        # Open for writing (flags are 0: read, 1: write)
  li   $a2, 0        # mode is ignored
  syscall            # open a file (file descriptor returned in $v0)
  move $s6, $v0      # save the file descriptor
  # Write to file just opened
  li   $v0, 15       # system call for write to file
  move $a0, $s6      # file descriptor
  la   $a1, outData   # address of buffer from which to write
  add   $a2,$zero,$t0       # hardcoded buffer length
  syscall            # write to file
  # Close the file
  li   $v0, 16       # system call for close file
  move $a0, $s6      # file descriptor to close
  syscall            # close file
  jr $ra
```

This procedure writes the outData into the input file. (Warning: The given input file will change after running the program)

`` capital:addi $a0,$a0,-32 jr $ra ```	This procedure makes given letter capital
`` strlen: addi $t0, $zero, -1 #initialize count to start with 1 for first character loop: lb $t1, 0($a0) #load the next character to t0 addi $a0, $a0, 1 #load increment string pointer addi $t0, $t0, 1 #increment count beqz $t1, exit #end loop if null character is reached j loop # return to top of loop exit: jr $ra ```	This procedure calculates the size of the given argument. ($a0) I used this procedure for calculate the size of the given text ,outData and text representations of any digits.
`` delete_newline: add $t0,$zero,$zero add $t1,$t1,$a0 loop2: lb $a3,0($t1) addi $t1, $t1, 1 bne $a3,$zero,loop2 # Search the NULL char code beq $t1, $a1,exit3 # Check whether the buffer was fully loaded subi $t1, $t1, 2 # Otherwise 'remove' the last character sb $0, 0($t1) # and put a NULL instead exit3:jr $ra ```	This procedure deletes the newline from given input.(input is taken from the user for the file name. Therefore it is important to delete the newline character.)

EXTRA THINGS

- I wrote the modified content into the file.
- I took file name as an input from the user.