Nadine Nam
student ID: 811151197

The Code File:

- Nam - revCaseMin_FinalVer.asm
- Nam - revCaseMin.pdf

A brief Summary of Project Implementation

Regarding the project 2 implementation, the user enters 30 characters, and the upper/lower case mixed characters are printed out in the reverse case. Also the project figures out the lowest byte in the reversed string.

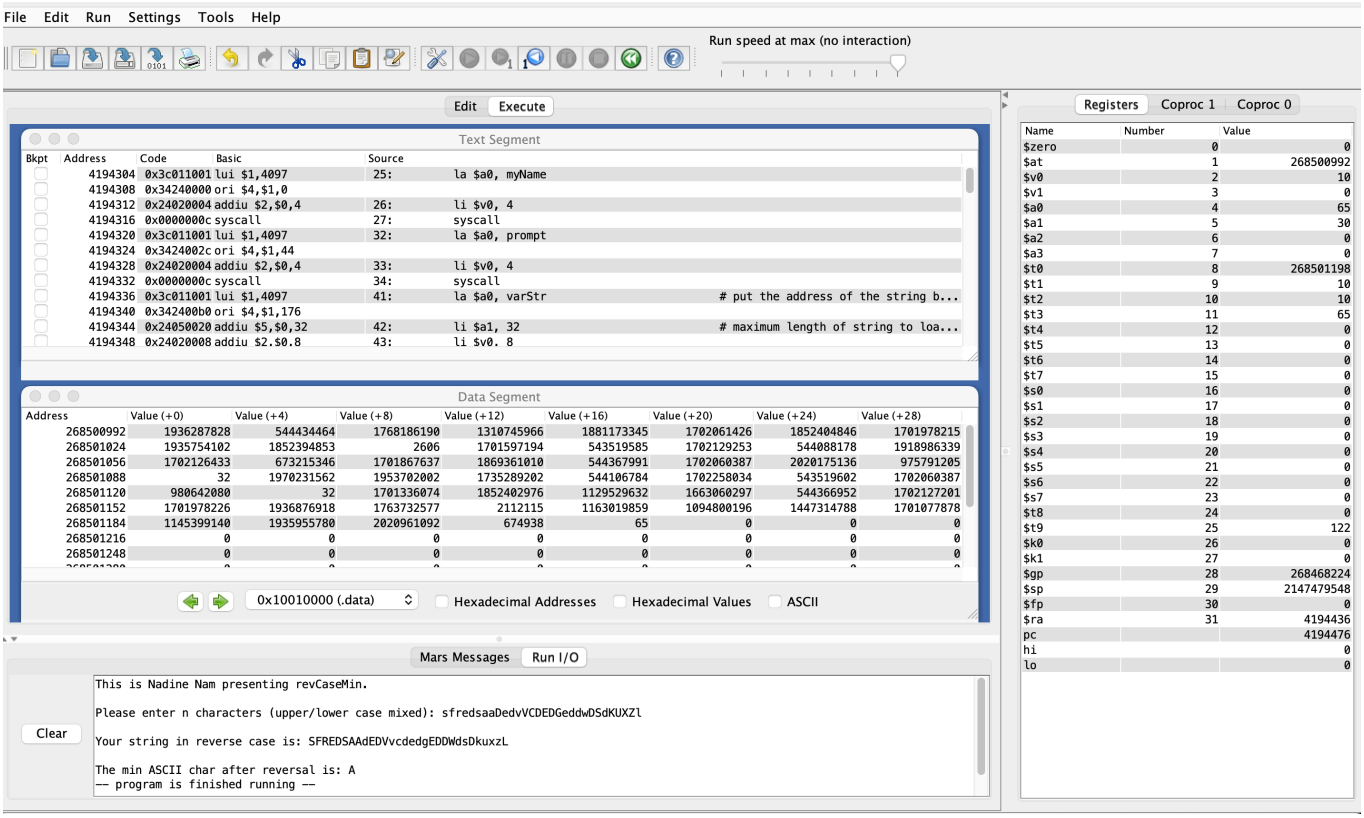
My project has overall three procedures. They are main(), revCase(), and findMin().
In the main(), it prints out prompt messages, inputs characters, and outputs the result of the reverse case and the min ASCII char.

For the Statement 1: revCase(), it has three sub-procedures. They are upperToLower(), lowerToUpper(), and exit().

For the Statement 2: findMin(), it has the same amount of three sub-procedures as well. They are loop_index(), update_minASCII(), and exit2().

Results showing the working code via screen prints

ENTIRE SCREEN



RUN I/O

The Conclusion listing the lessons learned and problems faced

Problems I faced

- At the first attempt of the assignment, I faced a problem that statement 2: minimum ASCII code wasn't printed out because the program didn't syscall its value in the end since a looping in findMin() didn't return the min ASCII char but zero byte of null character. Furthermore, program wasn't finished running as well.

1st Attempt (Error) : Nam - revCaseMin.asm

```
154  ################################ STATEMENT 2: findMin() ################################
155  FIND_MIN:
156          # <Load bytes at address $t0>
157          lb $t2, ($t0)
158
159
160          blt $t2, $t3, update_minASCII           # if ($t2 < $t3) { lowerToUpper }
161          bge $t2, $t3, loop_index
162
163          beq $t2, $zero, exit2
164
165          jr $ra
166
167
168          loop_index:
169                  sb $t2, ($t0)                   # store bytes in $t1 to $t0
170                  addi $t0, $t0, 1                # move to the next char
171                  # addi $t8, $t8, 1              # i++ (increment of array index)
172
173                  #bne $t6, $t7, FIND_MIN # if ($t0 != $t1) goto Loop
174                  j FIND_MIN
175
176
177          update_minASCII:
178                  move $t3, $t2
179
180                  j loop_index
181
182
183          exit2:
184                  sb $t2, minASCII
185
186                  jr $ra
187
```

```
Mars Messages  Run I/O

Assemble: assembling /Users/dnam/Library/CloudStorage/OneDrive-KentStateUniversity/Programming/CourseProjects/ComputerArchitecture

Assemble: operation completed successfully.

Go: running Nam - revCaseMin.asm

Error in /Users/dnam/Library/CloudStorage/OneDrive-KentStateUniversity/Programming/CourseProjects/ComputerArchitecture/Project2/su

Go: execution terminated with errors.
```

```
Mars Messages  Run I/O

This is Nadine Nam presenting revCaseMin.

Please enter n characters (upper/lower case mixed): abdbseDesDFFDWJDVDSdBLKHYHKFX
Your string in reverse case is: ABDBSEdESdffdwjdvdsDblkhyhkfx
```

The reversed string is printed out but the min ASCII char is not.
Got a message execution terminated with errors since the program wasn't finished.

I thought what was wrong with the first attempt file and why the program didn't end.

The reasons why the error occurred were

1. A $t3 register that saves the min ASCII character kept storing bytes even it went down to zero. The zero byte was actually from null character. Since the lowest bytes were 65('A'), a if-statement had to stop storing once if the value in $t3 register met 65.

2. After the loop_index(), the sub-procedure couldn't properly return the value to get out of FIND_MIN(). Thus my code caused the error that wasn't able to go back to main().

2nd Attempt (SUCCEED) : Nam - revCaseMin_FinalVer.asm

```
155  ################################ STATEMENT 2: findMin() ################################
156  FIND_MIN:
157          lb $t2, ($t0)                          # Load bytes at address $t0
158
159          blt $t2, $t3, update_minASCII          # if ($t2 < $t3) { lowerToUpper } -> $t3 = 122
160          bge $t2, $t3, loop_index               # if ($t2 >= $t3) { loop_index }
161
162
163
164          loop_index:
165                  sb $t2, ($t0)                  # store bytes in $t1 to $t0
166                  addi $t0, $t0, 1               # i++ (increment of array index), moving to the next char index
167
168                  j FIND_MIN                     # Looping: return to 'load bytes'
169
170
171          update_minASCII:
172                  blt $t2, 'A', exit2            # if $t2 is less than 65, then exits this FIND_MIN function
173
174                  move $t3, $t2                  # if $t2 is greater than 65, then save the min ASCII in $t3
175
176                  j loop_index                   # return to looping
177
178
179          exit2:
180                  sb $t3, minASCII               # save the FINAL minimum ASCII bytes
181                  jr $ra                         # return the value of FIND_MIN
182
183
```

```
This is Nadine Nam presenting revCaseMin.

Please enter n characters (upper/lower case mixed): sfredsaaDedvVCDEDGeddwDSdKUXZl

Your string in reverse case is: SFREDSAAdEDVvcdedgEDDWdsDkuxzL

The min ASCII char after reversal is: A
-- program is finished running --
```

Solution:

1. remove codes in the FIND_MIN() procedure

```
beq $t2, $zero, exit2
jr $ra
```

2. add new codes in the sub-procedure of update_minASCII():

```
FIND_MIN()
        update_minASCII:
                blt $t2, 'A', exit2
```

Lessons

In the project 2, I learned

- How to input characters as a long character array storing them in the memory.
- How to calculate bytes through registers.
- Getting used to the use of Opcodes
- How to properly convert upper/lower case by adding or subtracting bytes according to an ASCII code chart.
- Using the procedures and return their value (jal NAME, jr $ra).