Assignment-1 Submission

Indian Institute of Technology, Delhi

COL216: Computer Architecture

1 Introduction

The overall assignment is writing merge-sort algorithm for a list of strings in ARM. I have used ARMSIM# - Version 2.0.1 for the same. The assignment is divided into 3 components, the 3rd of which is submitted on 3rd Feb, 2022.

2 Stage 3

2.1 Objective

The objective of this stage is to finally write the merge sort algorithm that given a list of strings, sorts them. Functions made in stage 1 and stage 2 for string comparison and merging two sorted lists of strings have been used.

Since the input strings could be large in length, no string has been copied or moved. I have only dealt with the pointers to the strings.

2.2 Assumptions

I have used Version 2.0.1 of ARMSIM# and Angel SWI Commands. No use of Legacy SWI commands has been made.

I have assumed that the maximum number of strings is 250, and the sum of lengths of all strings (including null characters, but excluding newline characters) is 16000. This assumptions have been only made for static memory allocation. They can be easily relaxed by making a minor change in the code, if needed.
I have also taken care of the wrong inputs. If you try to enter a character or a negative integer for the number of input elements in a list, the program halts and starts to give an error.

2.3 Logistics

There are a total of 7 files. The main file contains the input/output logic and is named "main.s". The logic of sorting the is in a callable file and is in "merge_sort.s". The remaining files include

1. StringMatcher.s :- File that contains the logic of comparing 2 strings. This is the same file that was submitted in Stage 2.
2. ramio.s :- This file is credited to Ramanuj Goel. This contains the logic of input and output of a number.
3. math.s :- This file contains the logic of division of 2 numbers, which has also been taken from ramio,s.
4. UsefulFunctions.s :- This file contains the logic of getting input and giving output from stdin/stdout. This file was taken from Moodle and some minor changes were made to suit my own needs. 5. merge.s :- File that contains the logic of merging two sorted lists. This is the same file that was submitted in Stage 3.

2.4 Contributions

The functions to take input and print output from stdin/stdout respectively have been taken from the UsefulFunctions.s file available on Moodle. Minor changes were made in the functions to accomodate the personal needs. The code to directly input a number from console and directly output a number to the console has been taken from Ramanuj Goel, after due permission from the professor in the lecture. No help of any other form has been taken.

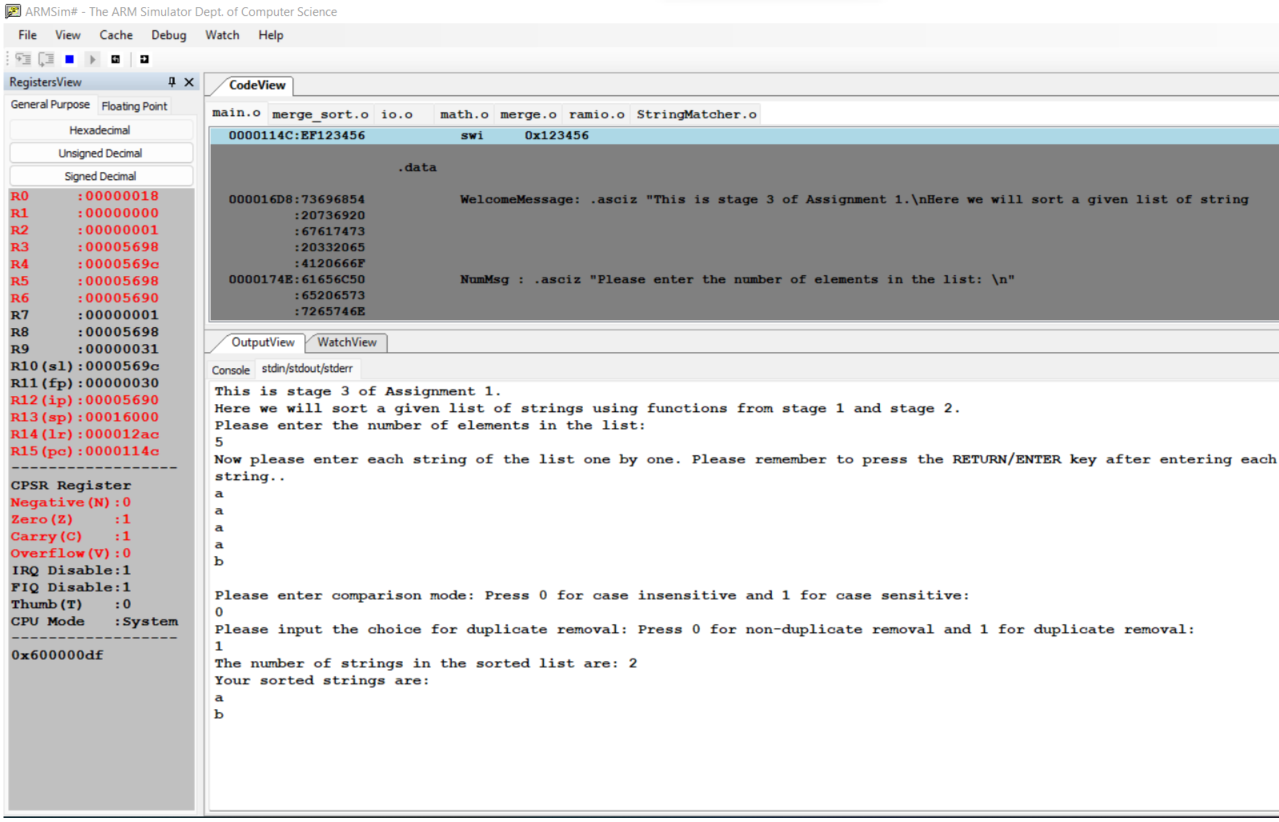
2.5 Test Cases

The program was tested on many different test cases.
My program first outputs the welcome message. It then expects the size of list which needs to be sorted. Once this has been given, the program will output message to get the inputs of strings of the list.

After all these inputs, the program expects to get 2 more inputs. The first input will be to get the comparison mode, and the second input will be to get the duplicate removal choice of the user,

This completes the set of inputs required. The program then outputs firstly the number of strings in the sorted list, and then outputs all the strings in sorted order, separated by new line characters.

The test cases were designed to exhaustively cover all the cases.

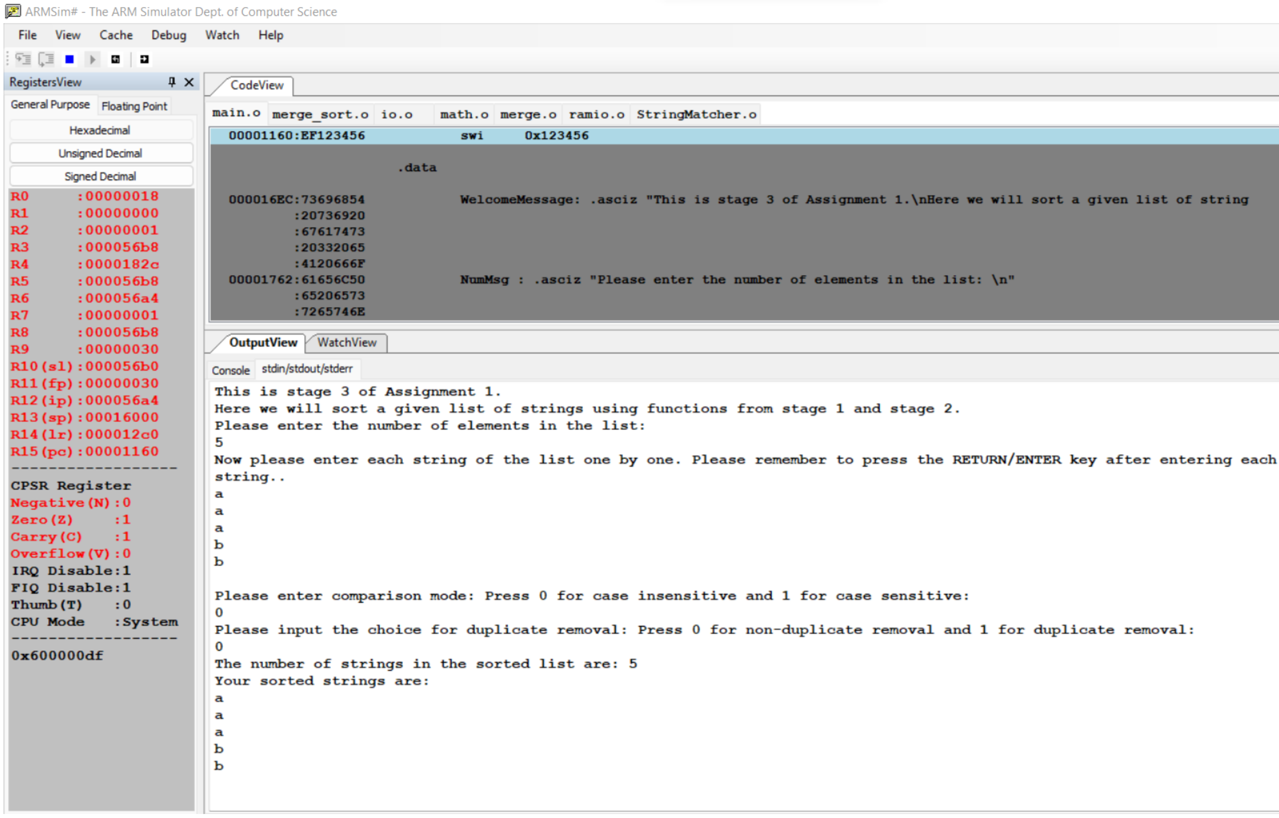


```
R0        :00000018
R1        :00000000
R2        :00000001
R3        :00005698
R4        :0000569c
R5        :00005698
R6        :00005690
R7        :00000001
R8        :00005698
R9        :00000031
R10(sl):0000569c
R11(fp):00000030
R12(ip):00005690
R13(sp):00016000
R14(lr):000012ac
R15(pc):0000114c
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
 0000114C:EF123456          swi      0x123456

                       .data

 000016D8:73696854          WelcomeMessage: .asciz "This is stage 3 of Assignment 1.\nHere we will sort a given list of string
         :20736920
         :67617473
         :20332065
         :4120666F
 0000174E:61656C50          NumMsg : .asciz "Please enter the number of elements in the list: \n"
         :65206573
         :7265746E
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
5
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
a
a
a
a
b

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
1
The number of strings in the sorted list are: 2
Your sorted strings are:
a
b
```

Figure 1: Test case 1

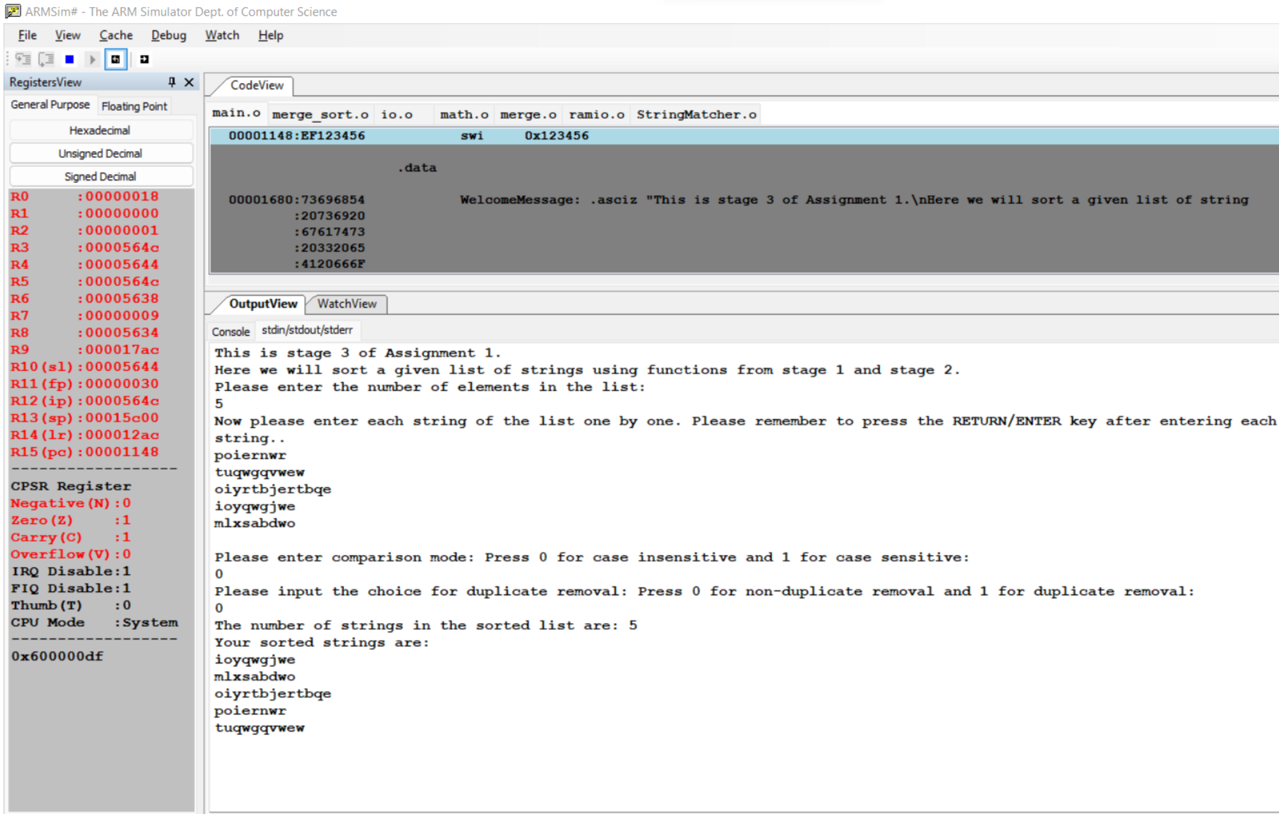


```
R0       :00000018
R1       :00000000
R2       :00000001
R3       :000056b8
R4       :0000182c
R5       :000056b8
R6       :000056a4
R7       :00000001
R8       :000056b8
R9       :00000030
R10(sl)  :000056b0
R11(fp)  :00000030
R12(ip)  :000056a4
R13(sp)  :00016000
R14(lr)  :000012c0
R15(pc)  :00001160
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
  00001160:EF123456          swi      0x123456

                     .data

  000016EC:73696854          WelcomeMessage: .asciz "This is stage 3 of Assignment 1.\nHere we will sort a given list of string
          :20736920
          :67617473
          :20332065
          :4120666F
  00001762:61656C50          NumMsg : .asciz "Please enter the number of elements in the list: \n"
          :65206573
          :7265746E
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
5
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
a
a
a
b
b

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
0
The number of strings in the sorted list are: 5
Your sorted strings are:
a
a
a
b
b
```

Figure 2: Test case 2

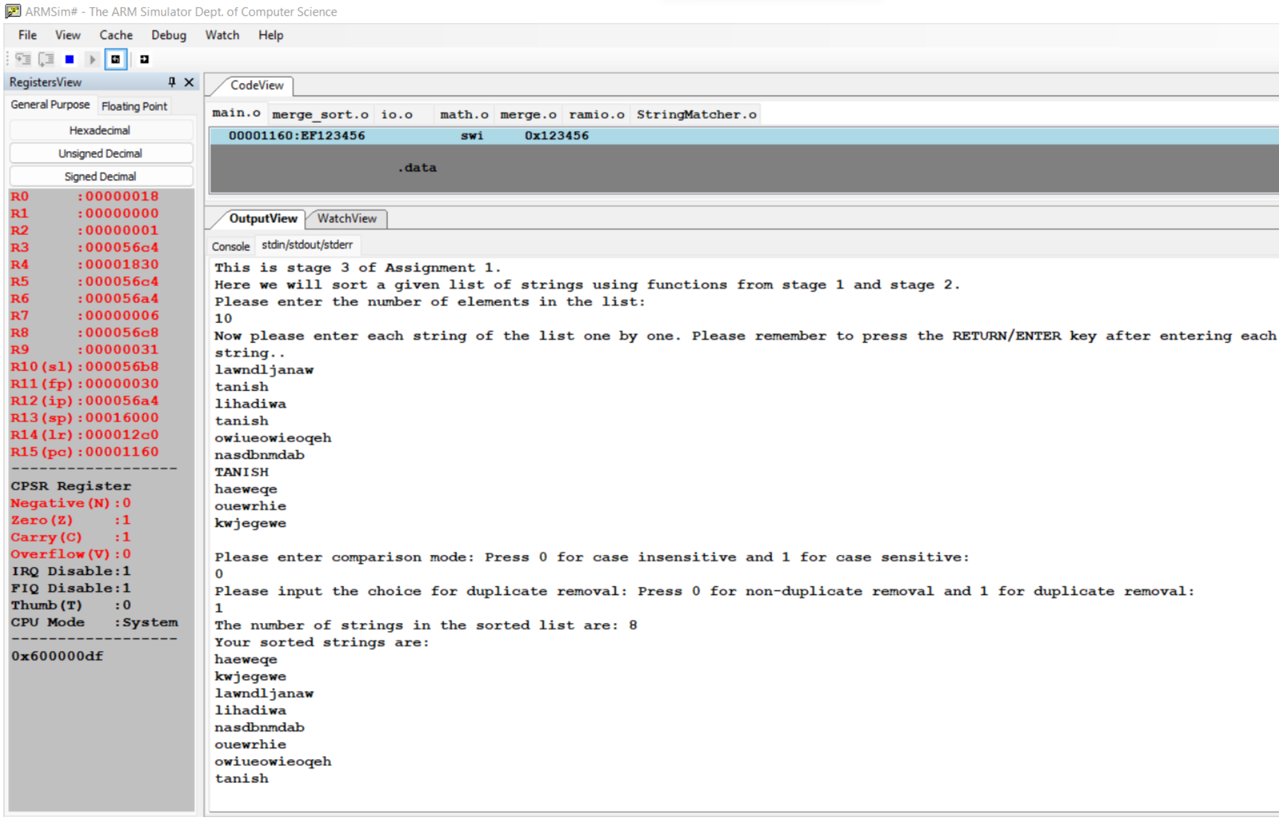


```
R0       :00000018
R1       :00000000
R2       :00000001
R3       :0000564c
R4       :00005644
R5       :0000564c
R6       :00005638
R7       :00000009
R8       :00005634
R9       :000017ac
R10(sl):00005644
R11(fp):00000030
R12(ip):0000564c
R13(sp):00015c00
R14(lr):000012ac
R15(pc):00001148
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
 00001148:EF123456          swi      0x123456

                     .data

 00001680:73696854          WelcomeMessage: .asciz "This is stage 3 of Assignment 1.\nHere we will sort a given list of string
         :20736920
         :67617473
         :20332065
         :4120666F
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
5
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
poiernwr
tuqwgqvwew
oiyrtbjertbqe
ioyqwgjwe
mlxsabdwo

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
0
The number of strings in the sorted list are: 5
Your sorted strings are:
ioyqwgjwe
mlxsabdwo
oiyrtbjertbqe
poiernwr
tuqwgqvwew
```

Figure 3: Test case 3

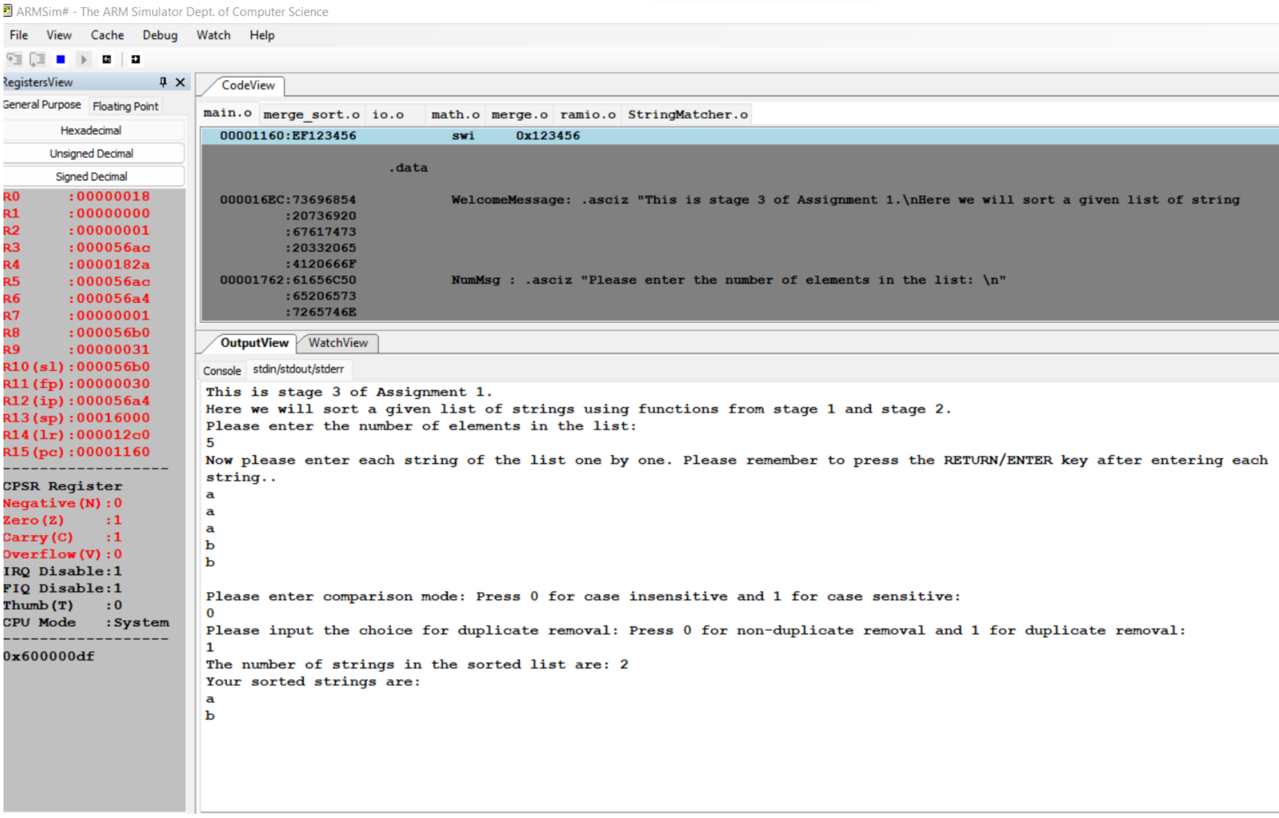

ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
CodeView
main.o merge_sort.o io.o math.o merge.o ramio.o StringMatcher.o
OutputView
WatchView
Console
stdin/stdout/stderr

```
R0        :00000018
R1        :00000000
R2        :00000001
R3        :000056c4
R4        :00001830
R5        :000056c4
R6        :000056a4
R7        :00000006
R8        :000056c8
R9        :00000031
R10(sl):000056b8
R11(fp):00000030
R12(ip):000056a4
R13(sp):00016000
R14(lr):000012c0
R15(pc):00001160
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
  00001160:EF123456          swi      0x123456

                      .data
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
10
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
lawndljanaw
tanish
lihadiwa
tanish
owiueowieoqeh
nasdbnmdab
TANISH
haeweqe
ouewrhie
kwjegewe

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
1
The number of strings in the sorted list are: 8
Your sorted strings are:
haeweqe
kwjegewe
lawndljanaw
lihadiwa
nasdbnmdab
ouewrhie
owiueowieoqeh
tanish
```

Figure 4: Test case 4

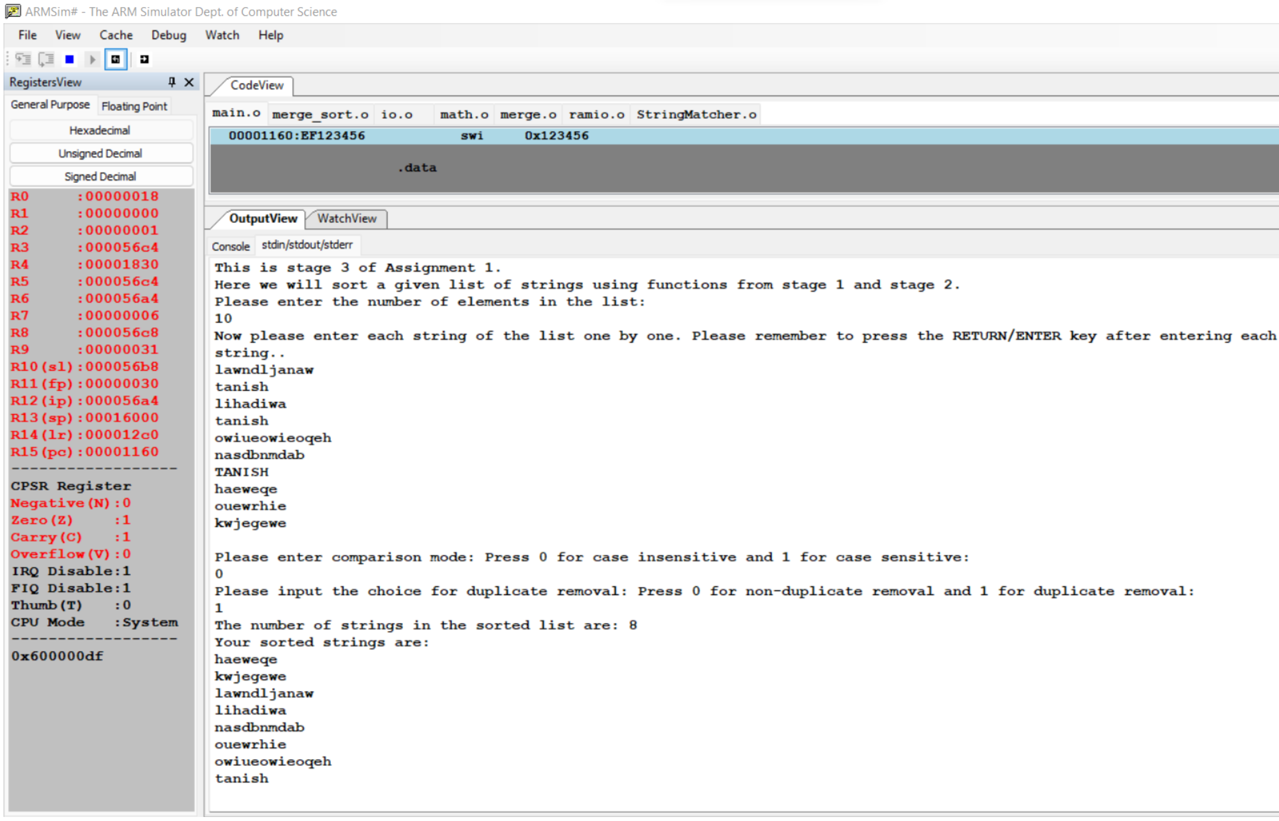


```
R0        :00000018
R1        :00000000
R2        :00000001
R3        :000056ac
R4        :0000182a
R5        :000056ac
R6        :000056a4
R7        :00000001
R8        :000056b0
R9        :00000031
R10(sl):000056b0
R11(fp):00000030
R12(ip):000056a4
R13(sp):00016000
R14(lr):000012c0
R15(pc):00001160
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
  00001160:EF123456            swi      0x123456

                        .data

  000016EC:73696854            WelcomeMessage: .asciz "This is stage 3 of Assignment 1.\nHere we will sort a given list of string
          :20736920
          :67617473
          :20332065
          :4120666F
  00001762:61656C50            NumMsg : .asciz "Please enter the number of elements in the list: \n"
          :65206573
          :7265746E
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
5
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
a
a
a
b
b

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
1
The number of strings in the sorted list are: 2
Your sorted strings are:
a
b
```

Figure 5: Test case 5



```
00001160:EF123456        swi      0x123456

                 .data
```

```
R0      :00000018
R1      :00000000
R2      :00000001
R3      :000056c4
R4      :00001830
R5      :000056c4
R6      :000056a4
R7      :00000006
R8      :000056c8
R9      :00000031
R10(sl):000056b8
R11(fp):00000030
R12(ip):000056a4
R13(sp):00016000
R14(lr):000012c0
R15(pc):00001160
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
This is stage 3 of Assignment 1.
Here we will sort a given list of strings using functions from stage 1 and stage 2.
Please enter the number of elements in the list:
10
Now please enter each string of the list one by one. Please remember to press the RETURN/ENTER key after entering each
string..
lawndljanaw
tanish
lihadiwa
tanish
owiueowieoqeh
nasdbnmdab
TANISH
haeweqe
ouewrhie
kwjegewe

Please enter comparison mode: Press 0 for case insensitive and 1 for case sensitive:
0
Please input the choice for duplicate removal: Press 0 for non-duplicate removal and 1 for duplicate removal:
1
The number of strings in the sorted list are: 8
Your sorted strings are:
haeweqe
kwjegewe
lawndljanaw
lihadiwa
nasdbnmdab
ouewrhie
owiueowieoqeh
tanish
```

Figure 6: Test case 6