

A4 (40 marks)

Objectives: MIPS Flow Control

Q1. Write a MIPS program that prompts the user for a hexadecimal digit and then displays the decimal equivalent multiplied by 2. For example, the program displays 10 for hex '5', 20 for 'A', etc. If the user enters an invalid value, the program should display an error message and asks the user to try again. (14 marks)

Hints:

- Valid characters can either be from '0' to '9' or from 'A' to 'F' (uppercase only).
- What is the hexadecimal value for ASCII '0' and 'A'?
- Shifting left by one bit doubles the value (if the new value fits in current register)

Sample run (running the program 3 times)

```
9
Hex digit X2 in decimal is 18
-- program is finished running (dropped off bottom) --

D
Hex digit X2 in decimal is 26
-- program is finished running (dropped off bottom) --

R
Unrecognized hex digit. Try again: R
Unrecognized hex digit. Try again: r
Unrecognized hex digit. Try again: d
Unrecognized hex digit. Try again: D
Hex digit X2 in decimal is 26
-- program is finished running (dropped off bottom) --
```

Q2. Write a MIPS program that prompts the user for a 4-bit binary number, reads it as a string, and outputs its decimal equivalent. When calculating the decimal value, use a loop to process the 4-character input string. Assume the user enters a valid binary number. (10 marks)

Hints:

- What is the hex decimal value for ASCII '0' and '1'?
- Shifting left by one bit doubles the value (if the new value fits in current register)
- New instruction: Shift left logical by a variable number of bits (not necessarily used here)
`sllv $s0, $s1, $s2` # \$s0 = \$s1 shifted left by the value in \$s2

Sample run

```
0101
5
-- program is finished running (dropped off bottom) --
```

Q3. Modify your code from Q2 so as to display the hexadecimal representation of the binary number instead of the decimal value. Assume the user enters a valid binary number. (6 marks)

Hints:

- You need to print a single *character* at the end: either '0' to '9' or 'A' to 'F'
- To convert from decimal to character representing a hex digit, add either 0x30 if the decimal value is from 0 to 9 or 0x37 if the decimal value is from 10 to 15.

Sample run (running the program 2 times)

```
0111
7
-- program is finished running (dropped off bottom) --

1110
E
-- program is finished running (dropped off bottom) --
```

Q4. Write a MIPS program that reads from the keyboard a string of no more than 20 characters and then outputs the string **in a single line** even if the input is written on multiple lines. The input string will be terminated by a character '0', which doesn't count as part of the string.

Hint: use a loop to read the input character-by-character and save all characters in a memory buffer. (10 marks)

Sample runs (green text is the output)

```
abc
def

ghi0
abcdefghi
```

Example of terminating input with 0

```
abcdef

ghijklm
nopq
abcdefghijklmno
```

*Example of reaching maximum
number of characters (20)*

Note: if any of your solutions is unreasonably long (e.g. twice as many instructions as another solution would need), then you might lose some marks for the extra instructions.

Submission Instructions: Compress all your files (including the assembly files) into one zip file and submit it **to Canvas**. Name your zip file as YourStudentID_A4.zip. You can resubmit an assignment, but the new submission overwrites the old one and receives a new timestamp.