# FORMAN CHRISTIAN COLLEGE

## (A CHARTERED UNIVERSITY)

### COMPUTER ORGANIZATION WITH ASSEMBLY LANGUAGE
### FINAL EXAM-1

**Maximum Marks: 100**

**Hard Deadline for Submission: Jun 30, before 11:59 pm**

**Weightage towards total marks: 20%**

**This homework alone carries 20% weightage of the total grade for this course. Marks obtained in this instrument will be contributed towards FINAL exam.**

**This home work is to be completed on group basis. A maximum of three students can form a group. Note that you must fill in the EXCEL sheet on google drive with the names and roll numbers of your group members. Once a group is formed it will not be changed/swapped.**

**A group can be formed of students from same section. No cross section group formation is allowed. If a group is found having members from both sections, they will get 50% cap on the marks obtained.**

**It's an open books and open notes task. Use of Internet is allowed, but you must provide references to web sites and /or tutorials from where you got help for this home work. You MUST NOT share your work with any of student/s from other group/s. Any such attempt if found will result in a ZERO grade. (Examiner has the right to test any or all documents for plagiarism and may call one or all groups for a viva). Note that if either of the components (code or report) is found plagiarized, all the students in the groups involved will get straight ZERO in FINAL EXAM.**

You need to submit code file/s along with a well formatted report. For each task in this assignment, your report should carry

- **Logic in bulleted points.**
- **Code**
- **Screen shots of all possible input / output.**

<u>**Rubric**</u>

| | |
|---|---|
| **Correct running code** | **80%** |
| **Well formatted report** | **20%** |

**Important Note:**

There are six programming exercises in this hand out. You must have a separate .asm file for each part. Every .asm file should start with a data dictionary as shown below:

**#Program Name: task1.asm**

**#Group Number: 20**

**#Group Members:**

| **#Name** | **Roll Number** |
|---|---|
| **#Rauf** | **123-456-78** |

**###############################**

# Task-1 [15 Marks]

In this task you need to write a program in MIPS assembly language that should get a string from user and compute the length of the given string. (Include all characters, printable as well as non-printable like spaces etc.)

**Sample Run 1**

```
Enter string:
```

raufbutt@fccollege.edu.pk

```
Length of string is: 25
```

**Sample Run 2**

```
Enter string:

COMP300 is a fun course!

Length of string is: 24
```

# Task-2 [10 Marks]

In this task you need to write a program in MIPS assembly language that should get a string from user and converts all the lower case letters in the string into upper case. Note that if there is one or more letters in the string that are already in upper case, your program should keep these intact. Further, if there are non-alpha characters in the string, the program should not alter these.

**Sample Run 1**

```
Enter string:
```

raufbutt@fccollege.edu.pk

```
Upper case version of the string is:
```

RAUFBUTT@FCCOLLEGE.EDU.PK

**Sample Run 2**

```
Enter string:

COMP300 is a fun course!

Upper case version of the string is:

COMP300 IS A FUN COURSE!
```

# Task-3 [10 Marks]

In this task you need to write a program in MIPS assembly language that should get a string from user and converts all the upper case letters in the string into lower case. Note that if there is one or more letters in the string that are already in lower case, your program should keep these intact. Further, if there are non-alpha characters in the string, the program should not alter these.

**Sample Run 1**

```
Enter string:
```

RAUFBUTT@FCCOLLEGE.EDU.PK

```
Lower case version of the string is:
```

raufbutt@fccollege.edu.pk

**Sample Run 2**

```
Enter string:

comp300 IS A FUN COURSE!

Lower case version of the string is:

comp300 is a fun course
```

# Task-4 [15 Marks]

In this task you need to write a program in MIPS assembly language that can simulate strncmp() function. Program should get two strings and an integer value n from user. It then compares first n characters of both strings, and displays the output. Output is 1 if the first n characters of both strings match and 0 otherwise. Note the characters to be matched include space/s also.

**Sample Run 1**

```
Enter first string:

COMP300 Computer Organization and Assembly Language

Enter second string:

COMP300 IS A FUN COURSE!

Enter number of characters to match:

4

Output is:

1
```

**Sample Run 2**

```
Enter first string:

COMP300 Computer Organization and Assembly Language

Enter second string:

COMP300 IS A FUN COURSE!

Enter number of characters to match:

10

Output is:

0
```

# Task-5 [15 Marks]

In this task you need to write a program in MIPS assembly language that can compute the frequency of occurrences of a character in a string. The input string should be provided by the user. User then provides the character. Your program should then process the string and outputs the frequency of the user given character in a specific format.

**Sample Run 1**

```
Enter string: ABDACBBBDDECEE
```

```
Enter character: a

Frequency of occurrence of given character: 0
```

**Sample Run 2**

```
Enter string: ABDACBBBDDECEE

Enter character: B

Frequency of occurrence of given character: 4
```

# Task-6 [15 Marks]

In this task you need to write a program in MIPS assembly language that can find and replace a character from within a string. User is prompted to enter the string. Your program then prompts user to enter the character to find. Finally it prompts user to enter the character to replace. Once all the information is provided, your program should find all occurrences of  first character in the string and replace it with the second character provided by the user.

**Sample Run 1**

```
Enter string: ABD ACBBB DDE CEE

Enter character to find : E

Enter character to replace: z

The find-n-replace results in:

ABD ACBBB DDz Czz
```

**Sample Run 2**

```
Enter string: ABD ACBBB DD ECEE

Enter character to find : Z

Enter character to replace: X

Could not found character Z in the string.
```

# Submission Format

You are required to submit following before the time window expires.

- Code file/s
- Report

Code file/s and report should be submitted in zip format with your group number as its name. Follow this naming convention:

  Group<your group number>_COMP300<your section>_SP_21_FINAL_EXAM_1

For example: **Group20_COMP300C_SP_21_FINAL_EXAM_1**

Make sure that the report should follow the template already shared with the class.