

## Introduction to Computing Laboratory

### Assignment – 8

---

1. Write a program in C to accept a word (string) through keyboard and convert all the characters in lower / upper case. Then check whether the word is palindrome or not.
2. Write a program in C that will accept a sentence as input with some extra spaces in between two words in the sentence. Then remove extra spaces such that only one space will remain in between any pair of words.
3. Write functions in C for the following purpose.
  - (a) To accept a string as argument and return the length of the string.
  - (b) To accept a string as argument and reverse the string.
  - (c) To concatenate two strings into a third one.
  - (d) To compare two strings character by character such that the function returns zero if the strings are similar. Otherwise the function returns the difference between the ASCII values of the two first-mismatching characters.
  - (e) To copy a string in another.In all the above cases you are to test the functions by calling them from main.
4. Write a program in C to accept a full name and convert it to initial mane as per the following examples. (*Sachin Ramesh Tendulkar* → *S. R. Tendulkar*, *Rahul Indra* → *R. Indra*)
5. Write a program to accept two numbers as strings, add them as integers and print the result as string. *Example: Accept "13" and "27" → 13+27=40 → print "40"*.
6. Write a program to accept 10 names in a two-dimensional array where each row represents a name (string). Now sort the names alphabetically. (*Note: you may also use an array of pointers to strings*).
7. Define a structure *student* above *main()* as follows:

```
struct student
{
    char name[50];
    int roll;
    float percentage;
};
```

Accept data for 10 such students in an array of structure and print them. Then print the record of the student with highest percentage of marks.
8. Define a structure to represent *complex* numbers. Write a function that will accept two complex numbers as arguments, add these two complex numbers and return the sum to the calling function.
9. Repeat the *problem 8* with the modification that the function will add two complex numbers as usual but not return anything. You are to call the function from *main()* and print the sum from *main()*.
10. Write a program to define a structure *number* as follows:

```
struct number
{
    struct complex comp;
    int real;
};
```

Here `struct complex` is same as *problem 8*. Populate such a structure with suitable values and print them.