

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
Second Semester 2017-18 CS F111 Computer Programming

LABORATORY SESSION #11

(Introducing Structures; more string manipulation)

1. In the lecture class, we discussed an example of a student structure that stores the different attributes of a student viz., name, ID number, age and CGPA.

```
struct student {  
    char name[20];    // stores only the first name  
    char id[14];  
    int age;  
    float cgpa;  
} s1;
```

- a. Define the structure in your program globally before the main() function, i.e., write the above few lines before main().
- b. Declare three variables of type struct student having the names s1, s2 and s3. Declare s2 globally, and s1 and s3 local to main().
- c. Inside main() find out how many bytes are allocated for s1.
- d. Accept user input for s1 and s2, using the dot operator for every member. For instance, to accept the name, you would write: scanf("%s", s1.name); and for scanning the CGPA, you would type: scanf("%f", &s1.cgpa); and so on.
- e. Assign the contents of s1 to s3 in one stroke.
- f. Now check if s2 is equal to s3, and print a message appropriately.
- g. Include the following function definition in your program:

```
void changeData(struct student s) { // receives a structure  
    strcpy(s.name, "Madhav");  
    s.age = 20;  
    s.cgpa = 7.33;  
}
```

Call this function from main() as follows:

```
changeData(s1);    // takes a structure as argument
```

Print the contents of s1 in main() before and after calling changeData().

What can you conclude about the mechanism of passing structures to functions?

- h. Print out the address of s3. Next, print out the addresses of the individual elements of s3. Note that the “member of operator” has the highest precedence among all operators. In other words, &s1.age will mean &(s1.age), even without the parenthesis.
- i. How would you declare a variable to store the address of s1?

2. Modify the function you wrote for question 10-3 (previous lab) so that a new function called **isAnagram2()** ignores the case of the letters of the strings that are compared. For example, this function would return a value 1 even for the two strings “Madam Curie” and “Radium came”. [Hint: You may use `toupper()` or `tolower()` whose declarations are available in `<ctype.h>` header file.]

A third version of this function, `isAnagram3()` that is not only case-insensitive, but also disregards any spaces found in the string has been implemented below, incompletely. (Note: A copy of this function, including `main()`, is available at: `/home/share/11.2.c` for you to copy into your directory.) For instance, this function is supposed to recognize each of the pairs such as “dormitory” and “dirty room”, “Slot Machines” and “Cash lost in me” as anagrams. The logic is to add up the ASCII values of each character of the strings (excluding space and tab) and compare the sums. If the sums are equal and so are the number of non-space characters of both strings, the strings are anagrams, else they are not!

```
int isAnagram3(char *w1, char *w2) {
    int i, sum_w1 = 0, sum_w2 = 0;
    for (i = 0; w1[i] ; ++i) {
        if (isspace(w1[i])) /* Disregard space and tab */
            /* >>>>>>>>> A statement goes here <<<<<<<<<< */
            sum_w1 += /* >>>>>>> An expression goes here <<<<<<< */ ;
    }
    for (i = 0; w2[i] ; ++i) {
        if (isspace(w2[i]))
            /* >>>>>>>>> A statement goes here <<<<<<<<<< */
            sum_w2 += /* >>>>>>> An expression goes here <<<<<<< */ ;
    }
    if (sum_w1 == sum_w2)
        return 1;
    else
        return 0;
}
```

- Complete the function and run the program.
- Does this logic work for all cases, or does it fail in any? Can you think of an example where it fails?
- (Homework) Can you design another algorithm for finding if two strings are anagrams? Hint: Think of how you would solve using pen and paper, and convert that into a C program.