BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI (RAJ.)

Second Semester 2017-18 CS F111 Computer Programming Practice Questions

(Topics: Dynamic Memory Allocation, Strings, Array of Pointers, Structures)

1. Write a function which: (a) takes a 1-D array with 10000 integers as input, (b) count the number of non-zero integers in it (let it be NZ number of integers), (c) declare another array of size NZ using dynamic memory allocation (using malloc() or calloc()), (d) returns the base address of this array.

2. Write a program which takes a string as input from the user. Let its maximum size be 10,000 characters. Ideally, there should be only one space (or tab) between any two words. But the string taken as input can have multiple such spaces or tabs. Your program should convert the string into another one with only one whitespace character between any two words. Write the following functions for it:
   a. Function getIdealLength(), which takes the string as an argument. It finds the number of additional spaces in the string and returns its ideal length. Note that ideal–length = original–length – number of additional spaces.
   b. Function createIdealString(), which takes the string and its ideal length as input parameters. The function then allocates the memory dynamically (of size ideal length), convert the string into another one with all additional spaces trimmed, and returns the base address of it.
   c. Call these functions in main().

3. Write a function that takes an input string and remove all the duplicate characters from the string. For example, if the input is “programming”, the output should be “progamin”.

4. Write a program to find the maximum occurring character in a given string (ignoring the case).

5. Write a C function **StoreSubstr**(**char *** str, **char** substr[][MAXCOL]) which takes a string **str** and computes all possible substrings (must be contiguous character(s)) of all lengths and stores them in the 2D array **substr**. Define MAXCOL using #define directive as the upper bound on the length of str.

6. Modify the above function so that **substr** is an array of pointers. Dynamically allocate memory for each of the substrings of **str** inside the called function.

7. The following details of students are made available in a file (as comma separated fields): ID number, name, gender (M or F), age (a whole number), residential status (H for hostel resident, D for day scholar), and CGPA. Two sample rows (i.e., two student records) of the data file (data.txt) are shown below:

   2014A7PS0108P,Indra Gopinath,F,21,H,8.55

   2017B1TS1055P,Mohammed Farhan,M,18,D,0.0

Problem statement: You will write a C program, made modular using user-defined functions to accomplish the following tasks:

a. Read each student record from the data file, and store in an array of structures, each element of the array representing a record.

b. Generate and store in the record the email address (BITS University email address) of each student.

c. Print out details of all students in the following format (one after another):

1. 2014A7PS0108P Indra Gopinath
Gender: F
Age: 21
Residence status: Hostel
CGPA: 8.55
Email: f2014108@pilani.bits-pilani.ac.in

2. 2017B1TS1055P Mohammed Farha
Gender: M
Age: 18
Residence status: Day scholar
CGPA: Not available
Email address: f20171055Žpilani.bits-pilani.ac.in

d. Calculate and print the average CGPA of all students whose CGPAs are available.

e. Calculate and print the number of: (i) male and female students, and (ii) hostel residents and day scholars

f. Print the CGPA of all the students sorted according to the ID numbers.

You will use the following structure definition in the program, globally:

```
typedef struct {
    char idno[13];
    char name[50];
    char gender;
    int age;
    char res_status;
    float cgpa;
    char emailaddress[40];
} STUD;
```

Some of the functions that can be used:

```
void populateRecords(STUDENT arr[]);
void generateEmail_address(char id[], char email[]);
void printRecords(STUDENT arr[]);
int calculateNumbers(STUDENT arr[],int gender, int res);
```