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

Second Semester 2017-18 CS F111 Computer Programming

LABORATORY SESSION #12

(Advanced Structures)

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
```

Write a C program, made modular using user-defined functions to accomplish the tasks listed below. But first, copy the file `/home/share/12.c` onto your current folder. The `main()` function with function calls, function prototypes as well as definitions are all there. The data file is available at `/home/share/12.data` you can use.

(a) Read each student record from the data file, and store in an array of structures, each element of the array representing a record. Write a function with the following prototype to do this task.

```
void populateRecords(STUD arr[], int no_studs);
                  /* function to get data into the array */
```

```
void populateRecords(STUD arr[], int n) {
   int i;
   for (i = 0; i < n; ++i)  {
     scanf("%[^,]", rec[i].idno);        /* scanning the first field */
     getchar();    /* dissipating the record delimiter character (,) */
       scanf("%[^,]", rec[i].name);
       getchar();
         rec[i].gender = getchar();  /* 'M' or 'F' */
         getchar();
            scanf("%d",&rec[i].age);
            getchar();
              rec[i].res_status = getchar();
              getchar();
                  scanf("%f", &rec[i].cgpa);
                  getchar();
    }
    return;
 }
```

(b) Generate and store in the record the email address (BITS University email address) of each student. The definition of the function for generating the email address, given the student ID number, has been provided to you.

```
void generateEmailAddress(char idno[], char email[]);
                  /* function definition already given */
```

Read the function definition and learn how it works, and also how to write meaningful comments for C code.

```
void generateEmailAddress(char idno[], char email[]) {
    char year[5];
    int i, yr, start, len, end;
    for (i = 0; i < 4; ++i)
      year[i] = idno[i];
    year[i]='\0';         /* year now contains the year of admission as
                               a string */
    strcpy(email,”f”);
    strcat(email,year);        /* email array now has fxxxx, where xxxx
                               is the year of admission */
    yr = atoi(year);    /* contents of year being converted into an int
                               */
    if (yr < 2017) /* prior to 2017 admissions */
      start = 9;    /* start extracting the last three digits in the ID
                               number*/
    else
      start = 8; /* start extracting the last four digits in the ID
                               number */
    len = strlen(email);
    end = 11;    /* extract till the last digit of the ID number */

    for (i = start; i <= end; ++i)
      email[len++] = idno[i];
    email[len] = '\0';

    strcat(email,"@pilani.bits-pilani.ac.in");    /* last part of email
                               address */
    return;
}
```

Now write the following function that calls `generateEmailAddress()` and then stores the email address in the right field of each student record:

```
void storeEmailAddresses(STUD arr[], int no_studs);
void storeEmailAddresses(STUD arr[], int n)
{
   int i;
   for (i = 0; i < n; ++i)
     generateEmailAddress(arr[i].idno, arr[i].emailaddress);
   return;
}
```

(c) Print out details of all students by writing a function whose prototype is:

```
void printRecords(STUD [], int);
```

Easy function to write – uses one loop inside the body of the function and prints out the data from the respective fields in the format that is asked for.

(d) Calculate and print the average CGPA of all CGPAs that are available:

```
float calculateAvgCG(STUD *, int);
```

Easy function to write. The only catch is that if the CGPA is 0.0, then it should not be taken into consideration (since the CGPA is deemed not available).

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

```
void printCount(STUD *, int gndr_sort, int resi_sort);
```

This function accomplishes the task based on how it is called:

```
printCount(arr,1,0); /* count students according to gender*/
printCount(arr,0,1); /* count according to resident status*/
```

Students may find it a bit difficult to understand how the same function can be used for both purposes. The body of the function should use an if or a switch statement to decide whether to count the males and females (second argument is 1), or to count the hostel residents and day scholars (third argument is 1), or do both (both the second and the third argument to the function is a 1).

(f) Print the CGPA of all the students sorted according to the ID numbers. It is not a good idea to sort the entire array of structures based on a specific field, but rather use a different strategy wherein the original array is left untouched.

This may be a challenging task for them to attempt. One way to approach this to use an array of pointers to structures. The pointers can be rearranged to reflect the sorted order of ID numbers. Then, one can iterate through the entire array of pointers and print the corresponding ID number and CGPA of students.