Data Structure Lab Assignment (CS 2172)
Assignment 1 Extension: Dynamic Memory Allocation concept
Time: 1 week

Problem

In this assignment, the proposed student-register mentioned in Assignment-1 should be implemented using the concept of dynamic memory allocation.

The typical structure definition for the same should be as follows

```
#define MAX_STUDENTS 10

typedef struct Student {
     char* name;     /* number of chars for name should be optimal. For
                     example, if the name of the Student is "Amlan Roy",
                     then a chunk of 10 charshould be allocated to name
                     field */
     int roll;
     char telephone[11];
     char * address;/* number of chars for address should be optimal.
                     For example, if the address of Amlan is "Flat 29,
                     Block 30, Newtown, Kolkata 700156", then a chunk of
                     43 charshould be allocated to address field */
} Student;

typedef struct SReg {

     int count;      /* Represents count of student entries, initially
                     itshould be set to 0. However, this is redundant
                     as it can be learnt by traversing arrStudPtr*/

    Student *arrStudPtr[MAX_STUDENTS];
                     /* The array of Student pointer that holds pointer
                     to student structure. */
} SReg;
```

A more flexible structure declaration should be as follows –

```
typedef struct SReg {
     int maxStuds;    /* Represents maximum number of possible students
                     in the register. It should be set based on user
                     input during runtime, not limited by MAX_STUDENTS*/

     int count;      /* Represents count of student entries, initially
                     it should be set to 0. However, this is redundant
                     as it can be learnt by traversing arrStudPtr */

    Student **arrStudPtr;
                     /* The array of Student pointer should be
                     allocated in runtime based on user input. */
} SReg;
```

The associated functions for the students-register should remain the same except some changes in the parameters. However, a few new functions also needed as stated below

- ***Student * createStudentRecord(…)*** – dynamically allocate memory for a Student record. Then reads respective fields of a Student record. For the pointer fields such as name and address, the memory should be allocated dynamically. Finally the functionreturns the student pointer. In case any failure of memory allocation, the function should return NULL.
- ***Student * createStudentRecordFromFile(…)*** – This function does the similar job as *createStudentRecord(...)*except it reads a file to populate a student record except user input.
- ***void freeStudentRecord(Student * sp)*** – frees the dynamically allocated memory for the Student record.

Implement the student register as a C program with all the above mentioned functionalities. Write a suitable main() function for demonstrating that your student register program supports all the features.