**STRUCTURE AND DYNAMIC MEMORY MANAGEMENT**

**1. Refer the question 1 solved in “Structure and function”. Extend the above program to read a number of records from the user as a single command line argument (each record is delimited by a semicolon and record fields are delimited by comma) and store in an array of structures.**

**Sample input and output are given below.**

**Input: “user1,90;user21,100, userABC,56,userX,40”;**

**Output:**

**No. of records: 4**

**Record 1:**

**Name:user1, Percentage:90**

**Record 2:**

**Name:user21, Percentage:100**

**Record 3:**

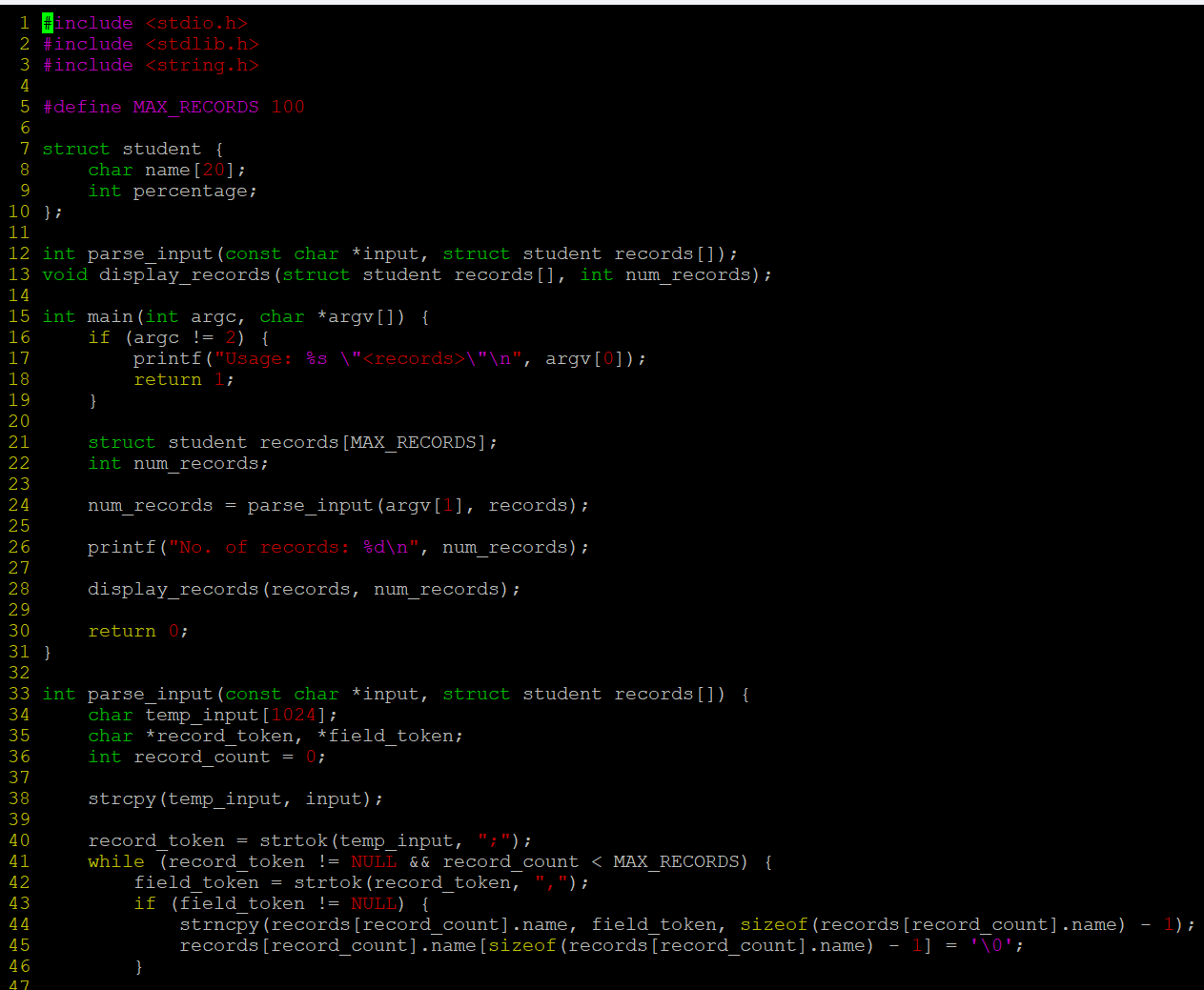
**Name:userABC, Percentage:56**

**Record 4:**

**Name:userX, Percentage:40**

**Implement all required functions and call them to get the desired output.**

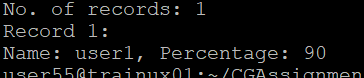
**Check for memory leak.**

****

**A computer screen with colorful text

Description automatically generated**

**OUTPUT:**

****

**2. Extend Q1. Above and add 3 functions below.**

**//to search for a name and to replace it with a user defined name, return replaced string**

**char\*search\_update(char \*searchstr, char \*replacestr);**

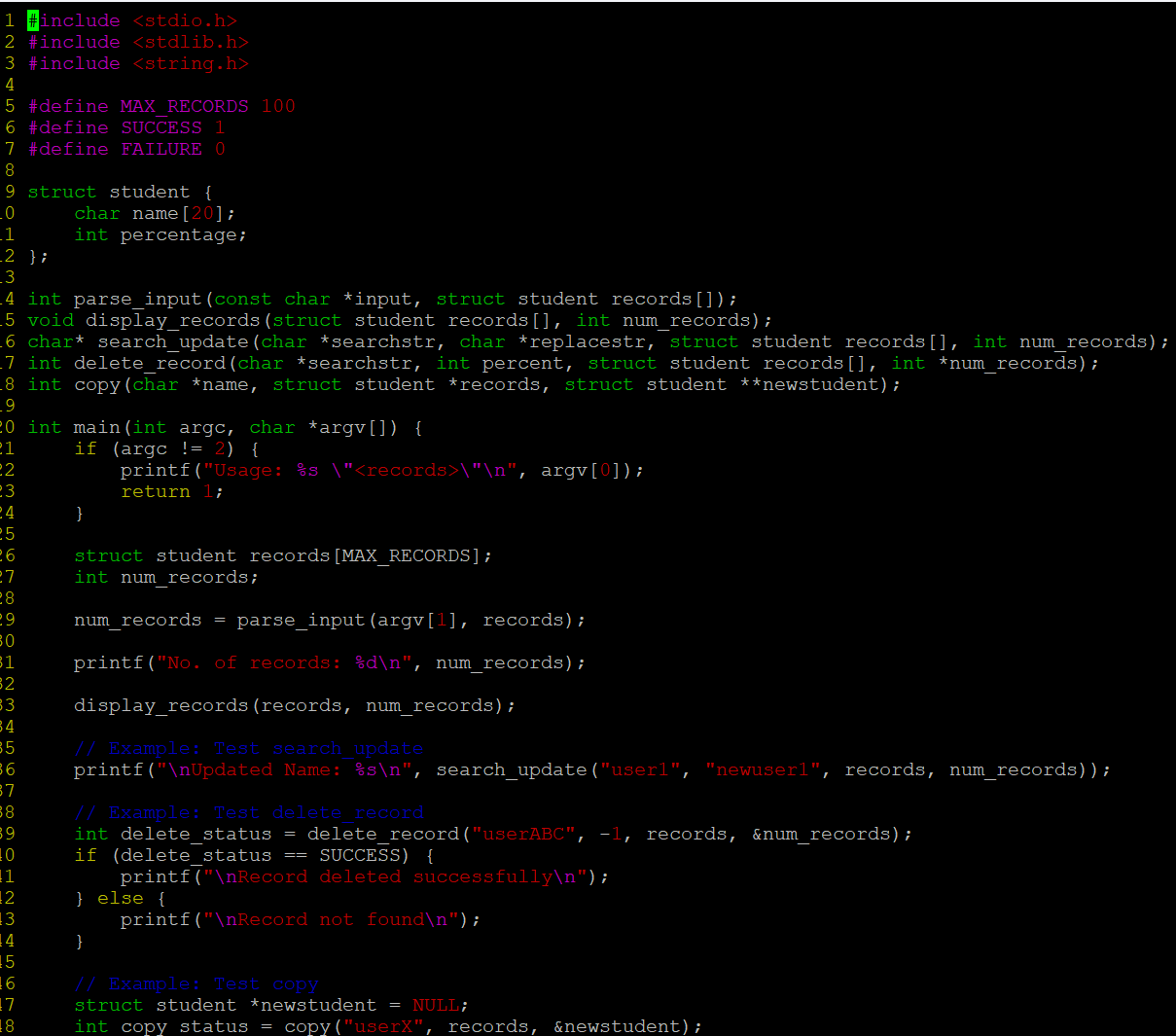
**//search and delete the record with given name or percentage value, return SUCCESS on successful delete else FAILURE**

**int delete\_record(char \*searchstr, int percent);**

**//search for name and if found create a copy of the record in newstudent**

**and return SUCCESS, else FAILURE**

**int copy(char \*name, struct student \*\*newstudent);**

****

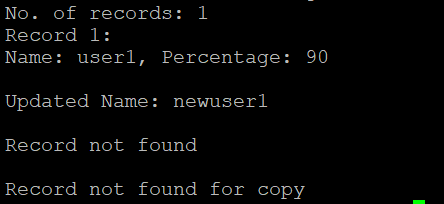
**A screen shot of a computer program

Description automatically generated**

**A computer screen with text on it

Description automatically generated**

**OUTPUT:**

****