CSE 115 Lab on Structure and functions – RRa

1. **Passing a Structure as function arguments (book records):**

```
#include <stdio.h>
#include <string.h>

struct Books {
  char  title[50];
  char  author[50];
  char  subject[100];
  int   book_id;
};

//function that takes a structure variable as a parameter
void printBook( struct Books book ) {
  printf( "Book title : %s\n", book.title);
  printf( "Book author : %s\n", book.author);
  printf( "Book subject : %s\n", book.subject);
  printf( "Book book_id : %d\n", book.book_id);
}

void main( )
{
  struct Books Book1, Book2;

  /* book 1 specification */
  strcpy( Book1.title, "C Programming");
  strcpy( Book1.author, "Nuha Ali");
  strcpy( Book1.subject, "C Programming Tutorial");
  Book1.book_id = 6495407;

  /* book 2 specification */
  strcpy( Book2.title, "Telecom Billing");
  strcpy( Book2.author, "Zara Ali");
  strcpy( Book2.subject, "Telecom Billing Tutorial");
  Book2.book_id = 6495700;

  printBook( Book1 );
  printBook( Book2 );
}
```

2. Passing an array of Structures as function arguments (book records):

```
#include <stdio.h>
#include <string.h>
#define MAX_BOOKS 1000

int NUM_BOOKS=0; //global variable containing the actual number of books

struct Books
{
   char  title[50];
   char  author[50];
   char  subject[100];
   int   book_id;
} ;

void readBooks( struct Books b[] )
{
   /* read book specifications from user user until s/he enters empty string as title*/
   int i;
   for(i=0; i < MAX_BOOKS; i++)    {
      printf("Enter book title (press just enter to finish): ");
      gets(b[i].title);
      if(strcmp(b[i].title, "")==0) break;
      printf("Enter author-names: ");
      gets(b[i].author);
      printf("Enter subject: ");
      gets(b[i].subject);
      printf("Enter id: ");
      scanf("%d", &b[i].book_id);
      fflush(stdin);
      NUM_BOOKS++; //update the number of books we have
   }
}

void printBooks( struct Books b[] )
{
   int i;
   printf("\n\n We have the following books:\n\n");
   for(i=0; i < NUM_BOOKS; i++)    {
      printf( "Book title : %s\n", b[i].title);
      printf( "Book author : %s\n", b[i].author);
      printf( "Book subject : %s\n", b[i].subject);
      printf( "Book book_id : %d\n\n", b[i].book_id);
   }
}
```

```
void main( )
{
   struct Books books[MAX_BOOKS];
   readBooks(books);
   printBooks( books );
}
```

Try yourself : Write a function called search that takes an array of Books structures and a string called title i.e. the header of the function will be: void search(struct Books b[], char title[])**. This function finds the book in the array b[] whose title is the same as the parameter called title and then prints all the info (title, authors, id, subject) of that book.**

Project Exercise:

1. Write the following functions for your project:
    a. view – for viewing records in your file,
    b. add – for adding records in your array of structures,
    c. search – this function will read a search key (e.g. student-id or name if your project is a student database) from user, search for that key in your array of structures, and then show info of all the records that matches the search key; for e.g. if the user searches for all students whose name is "Hafizur Rahman" then your program will show the info of all students whose name is "Hafizur Rahman"
    d. edit – this function will read a search key, show the info of all records which matches that key (just like search function), asks the user to select which record s/he wants to edit/modify, read the new info given by the user and use those to replace the old values of that record..
2. Create a menu in main function from which the user will choose whether s/he wants to view, add, edit, or search and call the corresponding function upon user input i.e., instead of adding a long piece of code under each case of your switch/case statement in main function (like you did in the previous class), call the respective function in each case.

Project Assignment:

1. Add the functions to delete a record from your array of structures.

Bonus: allow the user to do a partial search, *i.e.*, the user will be able to find all records which partially matches (instead of exact match) the name given by the user. For e.g. if anyone searches for "man" it will be able to show all records (e.g. students for student database) whose name contains "man" such as "manish", "rahman", "manik", "lokman", "amanat", etc.