CSE 115 Lab on Structure and functions – Ara2

1. **Structure variable declaration:**

```
struct person
{
    char name[50];
    int cit_no;
    float salary;
};

void main(){
 struct person p1, p2, p[20];
}
```

```
struct person
{
    char name[50];
    int cit_no;
    float salary;
}p1 ,p2 ,p[20];
```

2. **C Program that reads two distances (in feet+inches) and prints their sum:**

```
#include <stdio.h>
struct Distance{
   int feet;
   float inch;
}d1,d2,sum;

int main(){
   printf("1st distance\n");
   printf("Enter feet: ");
   scanf("%d",&d1.feet);  /* input of feet for structure variable d1 */
   printf("Enter inch: ");
   scanf("%f",&d1.inch);  /* input of inch for structure variable d1 */
   printf("2nd distance\n");
   printf("Enter feet: ");
   scanf("%d",&d2.feet);  /* input of feet for structure variable d2 */
   printf("Enter inch: ");
   scanf("%f",&d2.inch);  /* input of inch for structure variable d2 */

   sum.feet=d1.feet+d2.feet;
   sum.inch=d1.inch+d2.inch;

   if (sum.inch>12){  //If inch is greater than 12, changing it to feet.
      ++sum.feet;
      sum.inch=sum.inch-12;
   }
   printf("Sum of distances=%d\'-%.1f\"",sum.feet,sum.inch);
}
```

3. Array of structs (using 10 entries):

```
#include <stdio.h>
struct student{
   char name[50];
   int roll;
   float marks;
}s[10];
void main(){
   int i;
   printf("Enter information of students:\n");
   for(i=0;i<10;++i)
   {
      s[i].roll=i+1;
      printf("\nFor roll number %d\n",s[i].roll);
      printf("Enter name: ");
      scanf("%s",s[i].name);
      printf("Enter marks: ");
      scanf("%f",&s[i].marks);
      printf("\n");
   }
   printf("Displaying information of students:\n\n");
   for(i=0;i<10;++i)
   {
    printf("\nInformation for roll number %d:\n",i+1);
    printf("Name: ");
    puts(s[i].name);
    printf("Marks: %.1f",s[i].marks);
  }
}
```

4. 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 );
}
```

5. 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.**

EXERCISE:

1. **Create a struct called Student (see below) and read the records of two students using it. Then print the name and id of the student who has higher CGPA than the other.**

```
struct Student{
    char name[50];
    int id;
    float CGPA;
};
```

2. **Create a struct called BirthCertificate (see below) and read the info of two persons using it. Then print the name of the person who is older than the other. Also print his/her parents' names.**

```
struct BirthCertificate {
    int day, month, year; //to represent birthdate
    char name[100], fatherName[100], motherName[100];
};
```

3. **Create a struct called "Employee" with the following members:**
    a) **Name**
    b) **Date of Birth (D.O.B., in short)**
    c) **Starting Date**
    d) **Salary**

   Create an array of 4 employee variables; then read user input to fill up this array. Then display the info of the employee who gets the highest salary.

   Sample input/output:

   Name: **Bob Marley**
   D.O.B: **1/4/1993**
   Starting date: **12/6/2016**
   Salary: **20000**

   Name: **Rob Harfey**
   D.O.B: **2/11/1982**
   Starting date: **16/12/2016**
   Salary: **20000**

   Name: **katty Harley**
   D.O.B: **12/4/1993**
   Starting date: 2/6/2016
   Salary: 30000

   Name: **Betty Simpson**
   D.O.B: **3/11/1980**
   Starting date: **25/11/2016**
   Salary: **10000**

Info of employee with highest salary:
Name: katty Harley
D.O.B: 12/4/1993
Starting date: 2/6/2016
Salary: 30000

Hint: You can use `scanf("%d/%d/%d", &d, &m, &y);` command to read dates in the above format.

Bonus: Print the info of the employee who joined most recently (for the above inputs, the most recently joined employee is: Rob Harfey with starting date: 16/12/2016). You must create another struct for DOB and starting date (see nested structure). You can use the logic of comparing two dates in exercise 2 here.