CSE115L – Computing Concepts Lab
Lab 14

Structure declaration, initialization, accessing structure components and using `typedef`:

```c
#include<stdio.h>
#include<string.h>
struct Books
{
   char title[20];
   char author[20];
   char subject[20];
   int book_id;
};
int main()
{
   struct Books book1;
   strcpy(book1.title,"Programming");
   strcpy(book1.author,"Nuha Ali");
   strcpy(book1.subject,"Tutorial");
   book1.book_id=100;

   printf("%s\n",book1.title);
   printf("%s\n",book1.author);
   printf("%s\n",book1.subject);
   printf("%d\n",book1.book_id);
   return 0;
}
```

```c
#include<stdio.h>

typedef struct participant
{
   char name[20];
   char country[20];
   float score;
   int age;
}P;

int main()
{
   P player1={"David","Spain",9.65,25};

   printf("%s\n",player1.name);
   printf("%s\n",player1.country);
   printf("%.2f\n",player1.score);
   printf("%d\n",player1.age);

   return 0;
}
```

Array of structures, pointer to structure:

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

typedef struct participant
{
   char name[20];
   char country[20];
   float score;
   int age;
}P;

int main()
{
   P player[2];
   for(int i=0;i<2;i++)
   {
      gets(player[i].name);
      gets(player[i].country);
      scanf("%f",&player[i].score);
      scanf("%d",&player[i].age);
      getc(stdin);
   }
   for(int i=0;i<2;i++)
   {
      puts(player[i].name);
      puts(player[i].country);
      printf("%.2f\n",player[i].score);
      printf("%d\n",player[i].age);
   }

   return 0;
}
```

```c
#include<stdio.h>
void print(struct participant *pt,int size);
struct participant
{
   char name[20];
   char country[20];
   float score;
   int age;
};
int main()
{
   int n;
   scanf("%d",&n);
   getc(stdin);
   struct participant player[n];
   for(int i=0;i<n;i++)
   {
      gets(player[i].name);
      gets(player[i].country);
      scanf("%f",&player[i].score);
      scanf("%d",&player[i].age);
      getc(stdin);
   }
   print(player,n);
   return 0;
}
void print(struct participant *pt, int size)
{
   for(int i=0;i<size;i++)
   {
      puts(pt[i].name);
      puts(pt[i].country);
      printf("%.2f\n",pt[i].score);
      printf("%d\n",pt[i].age);
   }
}
```

Problems:

1. Create a structure called **`Shape`** which has two components, **`length`** and **`width`**. Create a structure variable **`Rectangle`** and take its length and width as input from the user. Implement the two functions **`int findArea(struct Shape R)`** and **`int findPerimeter(struct Shape R)`**. From the main function, call these two functions to get the area and perimeter of the rectangle.

```
struct Shape
{
        double length;
        double width;
};
```

2. Write a program to add two complex numbers using structure. Create a structure called **`Complex`** with two components, **`real`** and **`imag`**. Write a function that takes two structure variables as input, then sum up the two complex number.

```
struct Complex
{
        float real;
        float imag;
};

struct Complex add(struct Complex n1, struct complex n2);
```

3. Define a structure named **MovieStar** which will have the following elements*: Name (string), Rating(float), TotalFans(int)*. Declare a structure array of **MovieStar** for 5 movie stars. Now take N user reviews as input. Each review will consist of a Movie star name and his rating by a new fan. Now adjust each Movie Star's rating according to the reviews and show the results. Rating of a movie star is the average rating given by fans those who rated him.

4. Define a structure named **Gamer** which will have the following elements: *Number_of_ favorite_games (int), List_of_favorite_games (2D string)*. Now declare a structure array of **Gamer** for 5 gamers and take inputs for them. Now generate a rank list of the games. (Hint: The game which appeared most in the favorite games list will be the top game. In case of tie, print the game which comes alphabetically before).

5. Create a structure called **`BarcelonaPlayer`** with the following members.

```
struct BarcelonaPlayer
{
    char name[20];
    int age;
    char country[20];
    char Position[20];
    double Salary;
    double Rating;
};
```

First, create an array of **`BarcelonaPlayer`** structures. Now, write a function that takes an array of **`BarcelonaPlayer`** structures as input and find out the highest paid player among all the players.

```
void highestPaidPlayer(struct BarcelonaPlayer *pl, int size);
```

Create another function that finds all the players from Argentina.

```
void findPlayers(struct BarcelonaPlayer *pl, int size);
```