General Instructions

- Maintain a proper observation of the experiments you perform in lab sessions.
- Keep your code with proper hierarchical structure for your future reference.

Experiment 1

Write a C program that accepts two complex numbers u,v and stores as structures. Write individual functions (Must use functions and arguments using call by reference) for the following operations:

- Displays complex number
- computes u+v
- computes u-v
- computes u*v

Experiment 2

Write a C program that stores the data in an array of type "struct Employee". Data is given in the file *"emp.db"*.

Sample Input/Output

```
struct Employee{
int id;
char name[25];
struct Personal{
int age;
char gender;
};
struct Professional{
int date_of_joining;
char city[5];
int experience;
};
};
typedef struct Employee empData;
```

- Write a function that prints the employees' name who work in "Chennai".
- Write a function to calculate the average age of the employees in the company

Experiment 3

Run the given program and provide a brief explanation of how it works.

```
#include <stdio.h>
#include <string.h>
union Employee{
  float salary;
  int id;
  char name[5];
};
struct Employee_s{
  float salary;
  int id;
  char name[5];
```

```
};

int main() {
    union Employee data1;
    struct Employee_s data2;
    char name[] = "Groot";
   data1.salary = 123.12;
   data2.salary = 124.12;
   printf("Union  - %d,%f,%s\n",data1.id,data1.salary,data1.name);
   printf("Struct - %d,%f,%s\n",data2.id,data2.salary,data2.name);
   strcpy(data1.name,name);
   strcpy(data2.name,name);
   printf("Union  - %d,%f,%s\n",data1.id,data1.salary,data1.name);
   printf("Struct - %d,%f,%s\n",data2.id,data2.salary,data2.name);
   data1.id = 12;
   data2.id = 12;
   printf("Union  - %d,%f,%s\n",data1.id,data1.salary,data1.name);
   printf("Struct - %d,%f,%s\n",data2.id,data2.salary,data2.name);

   printf("Size of Union = %ld\n", sizeof(data1));
   printf("Size of Struct = %ld\n", sizeof(data2));
   return 0;
}
```