1. Matrix multiplication using 2D array:

```
void main()
{
    int A[100][100], B[100][100], C[100][100];
    int i, j, k, rowsA, columnsA, rowsB, columnsB;
    int sum;
    printf("Number of rows in A: ");
    scanf("%d",&rowsA);
    printf("Number of columns in A: ");
    scanf("%d",&columnsA);
    printf("Number of rows in B: ");
    scanf("%d",&rowsB);
    printf("Number of columns in B: ");
    scanf("%d",&columnsB);
    if(columnsA != rowsB) {
        printf("Invalid dimensions\n");
        return;
    }
    for(i=0;i<rowsA;i++)
    {
        for(j=0;j<columnsA;j++)
        {
            printf("A[%d][%d]: ",i, j);
            scanf("%d",&A[i][j]);
        }
    }

   for(i=0;i<rowsB;i++)
        {
            for(j=0;j<columnsB;j++)
            {
                printf("B[%d][%d]: ",i, j);
                scanf("%d",&B[i][j]);
            }

    }
    printf("Result:\n");
    for(i=0;i<rowsA;i++)
    {
        for(j=0;j<columnsB;j++)
        {
            printf("%10d ",C[i][j]);
        }
        printf("\n");
    }

}
```

2. 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);
  }
}
```

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.

Assignment:

1. **Declare the structure* needed for your project. Declare a 100 element array of that structure. Also declare an integer variable called num (with initial value zero) which indicates the number of *valid* (*i.e.,* elements read from user) elements in that array of structure variables. Declare the structure, array of structure, and num globally (outside all functions) and declare them before main function.**
   **** If you need more than one struct, you must declare them as well; for e.g. when you want to use a nested structure.***
2. **In main function, read the number of elements to be entered by user and assign that value to num (ask user to enter a value <= 100). Then read first num elements of your array of struct from user.**
3. **In main function, print the valid elements in your array of struct using a loop.**