# Laboratory 4

1. Questions
   1. Create a structure named company which has name, address, phone and noOfEmployee as member variables. Read name of company, its address, phone and noOfEmployee. Finally display these members’ value
   2. Create a structure TIME with members hours, minutes and seconds. Write a C program to add two time objects by passing structure variables to function and display result in H: M: S format.
   3. Write a program to read RollNo, Name, Address, Age & average-marks of ‘n’ students in a class. Print the list of students and the youngest and eldest student in the class.

1. Algorithm

**2.1 program to read name of company, its address, phone and No. Of Employee. Finally display these members’ value**

Step1: start

Step2: struct company{

char name[20];

char address[50];

long long int ph\_num;

int num\_em;

};

struct company company\_data;

Step3: void read\_data()

scanf("%s",&company\_data.name);

gets(company\_data.address);

scanf("%lld",&company\_data.ph\_num);

scanf("%d",&company\_data.num\_em);

Step4: void print\_data()

printf("\n\n Company Data:\n\n");

printf("Name of Company : %s \n",company\_data.name);

printf("Address of Company : %s\n",company\_data.address);

printf("Phone number of Company : +91 %lld\n",company\_data.ph\_num);

printf("Number of Employees in the Company%d\n",company\_data.num\_em);

Step5: stop

**2.2 program to add two time objects by passing structure variables to function and display result in H: M: S format.**

Step1: start

Step2: struct time

{int s,m,h;};

struct time T1, T2;

Step3: input data

printf("Enter the 1st time object\n");

scanf("%d%d%d",&T1.h,&T1.m,&T1.s);

printf("Enter the 2nd time object\n");

scanf("%d%d%d",&T2.h,&T2.m,&T2.s);

Step4: int min, sec, hr;

sec = T1.s + T2.s;

min = T1.m + T2.m + sec/60;

sec = sec%60;

hr = T1.h + T2.h + min/60;

min = min%60;

Step5: print added time

printf("Hours = %d, min = %d, sec = %d\n", hr,min,sec);

Step6: stop

**2.3 a program to read RollNo, Name, Address, Age & average-marks of ‘n’ students in a class. Print the list of students and the youngest and eldest student in the class.**

Step1: start

Step2: struct student{

int rl;

char name[10];

char address[20];

int age;

float marks;

};

struct student s[20];

Step3: input numbers of students, count

Step4: input data

scanf("%d%s%s%d%f",&s[i].rl,&s[i].name,&s[i].address,&s[i].age,&s[i].marks);

Step5: output data

printf("\nRoll no: %d name: %s address: %s age: %d marks: %f",s[i].rl,s[i].name,s[i].address,s[i].age,s[i].marks);

Step6: for calculating eldest and oldest student of the class

int old=s[0].age;

int eld=s[0].age;

for ( int i = 1; i < n ; i++)

{ if (s[i].age > old){

old=s[i].age;

o=i; }

if (s[i].age < eld){

eld=s[i].age;

e=i; }

}

Step7: print eldest and oldest student details

printf("\nEldest student: ");

printf("\nRoll no: %d name: %s address: %s age: %d marks: %f",s[e].rl,s[e].name,s[e].address,s[e].age,s[e].marks);

printf("\nOldest student: ");

printf("\nRoll no: %d name: %s address: %s age: %d marks: %f",s[o].rl,s[o].name,s[o].address,s[o].age,s[o].marks);

Step8: stop

1. Program



Figure 1 program to read name of company, its address, phone and No. Of Employee. Finally display these members’ value



Figure program to add two time objects by passing structure variables to function and display result in H: M: S format.



Figure   
 a program to read RollNo, Name, Address, Age & average-marks of ‘n’ students in a class. Print the list of students and the youngest and eldest student in the class.

1. Presentation of Results

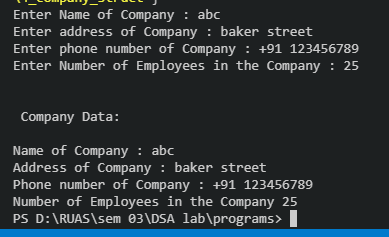


Figure output of program to read name of company, its address, phone and No. Of Employee. Finally display these members’ value

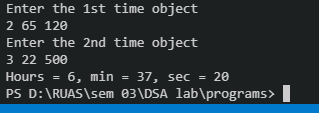


Figure output of program to add two time objects by passing structure variables to function and display result in H: M: S format.

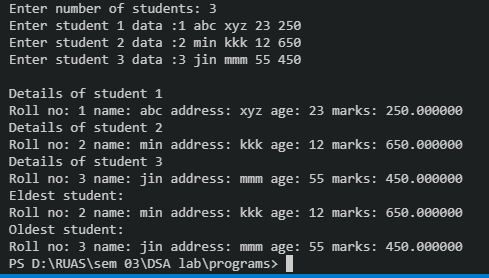


Figure ouput of a program to read RollNo, Name, Address, Age & average-marks of ‘n’ students in a class. Print the list of students and the youngest and eldest student in the class

1. Conclusions

Learning Happened:

* A structure is a user defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type
* ‘struct’ keyword is used to create a structure.Eg.,

struct address

{

   char name[50];

   char street[100];

   char city[50];

   char state[20];

   int pin;

};

* A structure variable can either be declared with structure declaration or as a separate declaration like basic types.
* Structure members are accessed using dot (.) operator.

Hence we can see the programs are compiled successfully without any error.