Structures in C

Dr. Chénche

Structure in C

Typedef :

The **Typedef** (short for "type definition") is an instruction allows to define a new name (alias) for an existing type.

Example :

```
typedef float real ;

int main(){
real x;
return 0;}
```

Structure in C

- C allows to group a set of variables together into a single item know as structure.

- A structure is a set of diverse types of date grouped together under a unique declaration.

For example, a student might keep the variables shown in the

all the variables listed in this table are related because they hold data about the same student;

Student	Data
int code;	student registration number
char name[20];	student name
int age;	student age
float average;	student average

Structure in C

Declaration

```
Syntax :
typedef struct {
    // Variable declarations go here
} name_of_structure ;  // Notice the required semi-colon
```

- **The name of structure** is used like a **data type name**.

- The variable declarations that appear inside the braces declare the **members of the structure**.

Structure in C

Example 01: Structure student

```
typedef struct {
    int code,age;
    char name[20]; float average;
} Student; // Here, 'Student' is the alias for this structure type
```

Example 02: Structure time

```
typedef struct {
    int hour,minute,second;
} time;
```

Structure in C

Typedef :

The **Typedef** (short for "type definition") is an instruction allows to define a new name (alias) for an existing type.

Example :

```
typedef float real ;

int main(){
real x;
return 0;}
```

Structure in C

Declaring a Structured Variable:

```
typedef struct {
  int hour,minute,second;
} time;

int main(){
time T;
return 0;}
```

This means that **T** is a variable of type **time**, and it will have three fields: hour, minute, and second.

Structure in C

Accessing the Fields of a Structured Variable:

We can access the members of a structure using the dot (.) operator:

```
int main () {
    time T;
 T.hour= 23; T.minute= 50; T.second= 36;
            }
```

Structure in C

Array of Structure:

It is possible to create an array of structures:

```
int main ()
{
time T[100]; // Array of 100 elements of type time
T[0].hour = 12; // Assign 12 hours to the first element of the array
}
```

Structure in C++

Matrix of Structure:

It is possible to create an array of structures:

```
int main ()
{
time M[100][100]; // Matrix of 100x100 elements of type Time
M[1][0].hour = 12; // Assign 12 hours to the element M[0][0]
}
```

Exercise 1 : (Course)

Let the structure "**Contact**" be defined as follows:

- Name: string of length ≤ **50**,
- Address: string of length ≤ **150**,
- Phone number: string of length ≤ **15**,
- Age: integer.

Write a program that allows you to:

1) Define the structure.

2) Read information of **n** persons (**n≤100**).

3) Display information of youngest person.

4) Show the name and age of all persons whose names exceed **30** letters (name length > **30**).

5) Display the name and address of all persons with a phone number that starts with "**021**" (for example: the number "**021365502**" starts with "**021**").

Exercise 2 : (Course)

Write a program in which you will :

1- Define a structure "**Employee**" which includes the following fields: Name, date of birth, function, date of recruitment, salary.

2- Enter the information of n given employees ($n \leq 100$),

3- Display the names of employees with a salary $\leq$ 30 000 da.

4- Display the information of the employee with the highest salary.

5- Display the salary mass of employees.

6- Display the name of the youngest employee.