ENUMERATED DATA TYPES

- Enumerated data type consists of named integer constant as a list.
- Enum Data Types starts with'0' by default and incremented by 1 in a sequential order
- It is a User defined Datatype
- It gives numbers to the name.

Syntax:

Enum identifier[optional{Enumerator_list}];

where, Enum-Keyword

Example:

Enum month{Jan, Feb, Mar} /*Jan, Feb, and March variables assigned to 0,1,2 by default.

Enum month{Jan=20,Feb,Mar} /*Jan, Feb, and March variables assigned to 0,1,2 by default.

Program:

```
#include<stdio.h>
int main()
{
enum MONTH{Jan=0, Feb, Mar};
Enum MONTH Month=Mar;
if(Month==0)
printf("value of Jan");
elseif(month==1)
printf("month is Feb");
if(month==2)
printf("Month is Mar");
return 0;
}
```

Output:

```
Month is March.
```

- The use of enum in c to name the integer values makes the entire program easy to learn, understand and maintain by the same or even different programmer.

Syntax to define Enum in C

- An enum is defined by using the "enum" keyword in c, and the use of a comma seperates the constants. The basic syntax of defining an enum is:

Enum enum_name{int_const1,int_const2, int_const3,….int_const N};

- In the above syntax, the default value of int_const1 is 0, int_const2 is 1, int_const3 is 2, and so on.
- However, you can also change these default values while declaring the enum.
- Below is an example of an enum named cars and how you can change the default values

  Enum cars {BMW, Ferrari, Jeep, Mercedes Benz};

  Here the default values for the constants are:

  BMW=0, Ferrari=1, Jeep=2 and Mercades Benz =3.

  However, to change the default values, you can define the enum as follows:

  Enum cars{BMW=3, Ferrari=5, Jeep=0, Mercedes_benz=1};

Enumerated Type Declaration to create a variable

- Similar to predefined data types like int and char. You can also declare a variable for enum and other user- defined data types. Here's how to create a variable for enum.

  Enum condition(true,false); //declaring the enum

  Enum condition e; //Creating a variable of type condition

- Suppose we have declared an enum type named condition; we can create a variable for that data type as mentioned above.
- We can also converse both the statements and write them as:

  Enum condition(true,false) e;

- For the above statement, the default value for true will be 1, and that for false will be 0.

How to create and implement Enum in C program

Example 1: Printing the values of weekdays

```
#include<stdio.h>
Enum days{Sunday=1, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday};
int main()
{
for(int i= Sunday; i<=Saturday;i++)
{
printf("%d",i);
}
return 0;
}
```

Output

```
1,2,3,4,5,6,7
```

- In the above code, we declared an enum named days consisting of the weekdays starting from Sunday.
- We then initialized the value of Sunday to be 1. This will assign the value for the other days as the previous values plus 1.

- To iterate through the enum and print the values of each day, we have created a for loop and initialized the value for i as sunday

How to use Enum in C?

- We use enums for constants ie, when we want a variable to have only a specific set of values.
- For instance, for weekdays enum, there can be only seven values as there are only seven days in a week.
- However, a variable can store only one value at a time.
- We can use enums in c for multiple purposes; Some of the use of enums are,
    1. To store constant values(eg. Week days, Months, Directins, Colors in a rainbow)
    2. For using flags in c
    3. While using switch case statements in c

Examples of Using Enum in Switch case statements

- In this example, we will create an enum with all the 4 directions, North, East, West and South as the constants.
- We will then use the switch case statements to swtich between the direction elements and print the output based on the values of the variable for the enum directions.

```
#include<stdio.h>
Enum directions{North=1,East, West, South};
int main()
{
enum directions d;
d=west;
switch(d)
{
case North:
printf("We are headed towards North");
break;
case East:
printf("We are headed towards East");
break;
case West:
printf("We are headed towards West");
break;
case South:
printf("We are headed towards South");
break;
}
return 0;
}
```

Output

We are headed towards West.

Example of using Enum in C for flags.

- We can use enum in c for flags by keeping the values of integral constants a power of 2
- This will allow us to choose and combine two or more flags without overlapping with the help of the Bitwise OR(1) operator.
- Let’s consider the example below where we set three flags: Crop, Rotate and save to work with an image.

Example

```
#include<stdio.h>
Enum designFlags
{
CROP=1, ROTATE=2,SAVE=4};
int main()
{
int my Example=ROTATE/SAVE;
printf(“%d”,myExample);
return 0;
}
```

Output

```
6
```