Function In C Programming:

In every programming language function plays an important role. Basically function is a block of code which has some name of identification.

Even in the C language the smallest program there is at least one function. All function names must be unique. In C language the main() function is the mandatory function, because compiler will start its execution from main() function.

So, that there is one function name, just be main() function(). You can define function is any sequence, No keyword is a function. Operating System calls main() function to begin the execution of every C programming source code.

Types Of Function:

Generally there are two types of functions are there in every programming language. In C programming language also there are two types of function is there

1. Pre-defined Function
2. User defined Function

Pre-Define Function:

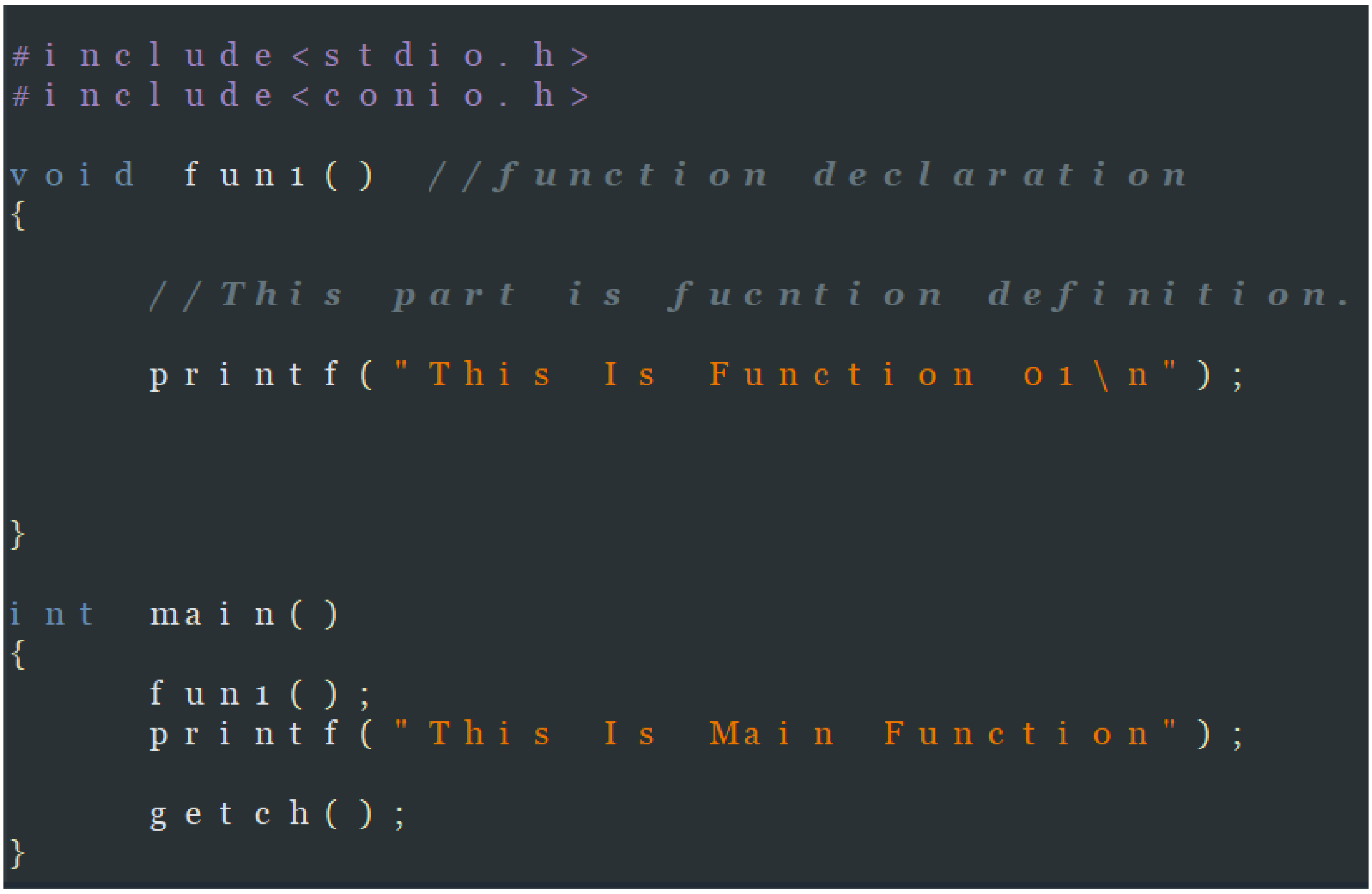
These are the types of function which is already created for C programming language to simplify the source code. Basically these functions actual code was stored in library files or you can say the header files, which are the (.h) extension files. Because of that in every source code we need to add the header file to simplify our source code. Example: printf(), scanf(), getch(), exit(), pow(), etc.

User Define Function:

These are the types of function which are created by user/programmer who wants to simplify the code according to its requirement. Function helps a user from so many ends. User defined function must be declared by programmer or user. The programmer has to write the code for such function and test them properly before using them. Example: Arithmetic(), Multiplication(), etc.

Function Definition & Function Declaration:

Function definition and declaration are the very easy and important concept in every programming language. Function Definition defines the body structure of a particular function. But the function declaration defines the identity of a particular function. Function declaration means function’s name, return type and parameters are the part of the function declaration.



```c
#include<stdio.h>
#include<conio.h>

void fun1() //function declaration
{

    //This part is fucntion definition.

    printf("This Is Function 01\n");


}

int main()
{
    fun1();
    printf("This Is Main Function");

    getch();
}
```

Parameter & Argument:

The variables declared in the definition of a function are referred to as parameters. They serve as placeholders or containers that hold values that are passed in from calling functions. The number of values and data type that the function expects as well as its interface (or "signature") is defined through parameters.

The actual values or expressions provided to a function during its invocation are known as arguments. They match the parameters set forth in the function and offer the actual information or values that are necessary for the function to carry out the tasks for which it was designed.

Writing modular, reusable, and effective code requires an awareness of the differences between parameters and arguments. Programmers may write functions that interact with other areas of their codebase with ease by understanding how parameters and arguments work together.

Advantage Of Function:

1. Same code need not be written again. So length of the source code decreases.

2. As the program length decreases the memory required for the program also decrease.

3. We can call function any number of times from any place.

4. As the program length decreases time taken less for write code.

Large library function that carry out commonly used calculation and operation available along with C language. In order to use these function we need to include them into our program. these functions are stored in special file called the header files. Each header file contains information about the group of library functions.

Return Keyword:

In a function the “return” statement returns the flow of the execution to the function from where it is called. The return keyword finished the execution of a method, and can be used to return a value from a method.

Way To Define A Function:

Generally to define a function there are four ways are there. They are as follows:

1. Take Nothing Return Nothing (TNRN)

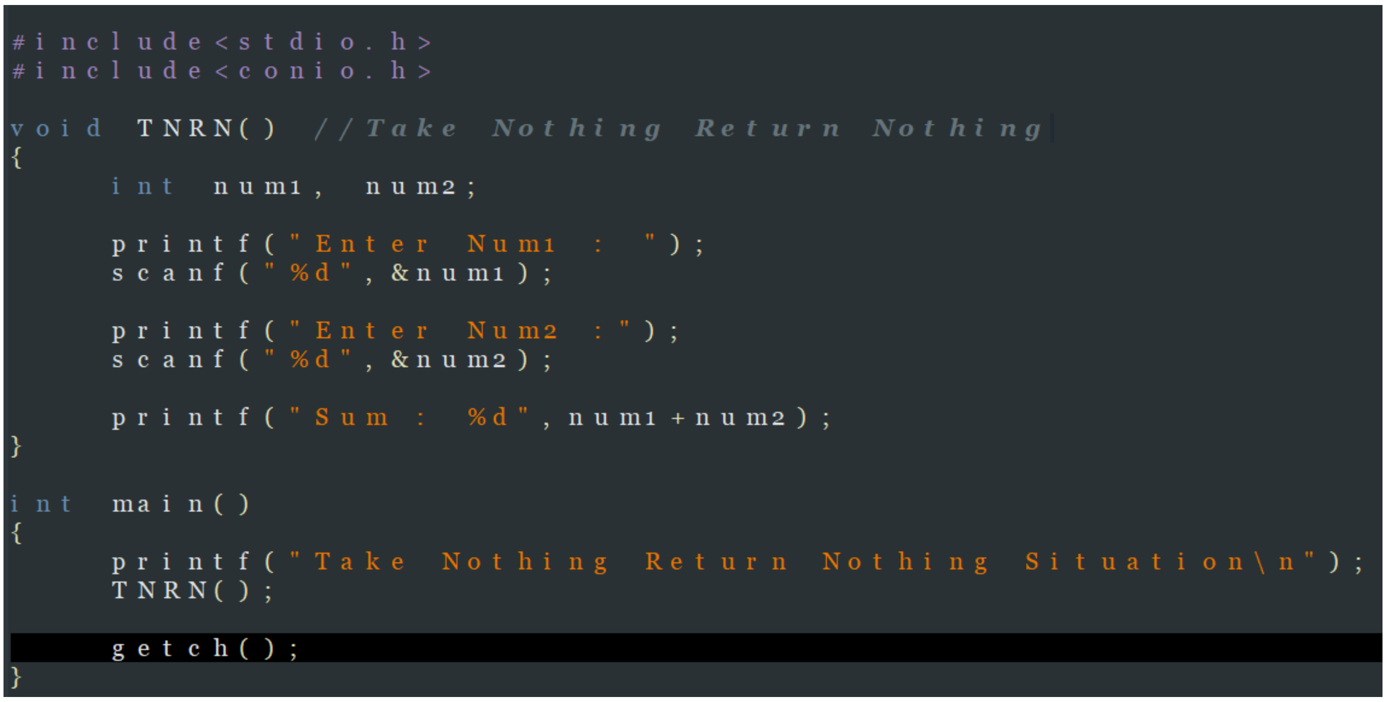
2. Take Something Return Nothing (TSRN)

3. Take Nothing Return Something (TNRS)

4. Take Something Return Something (TSRS)

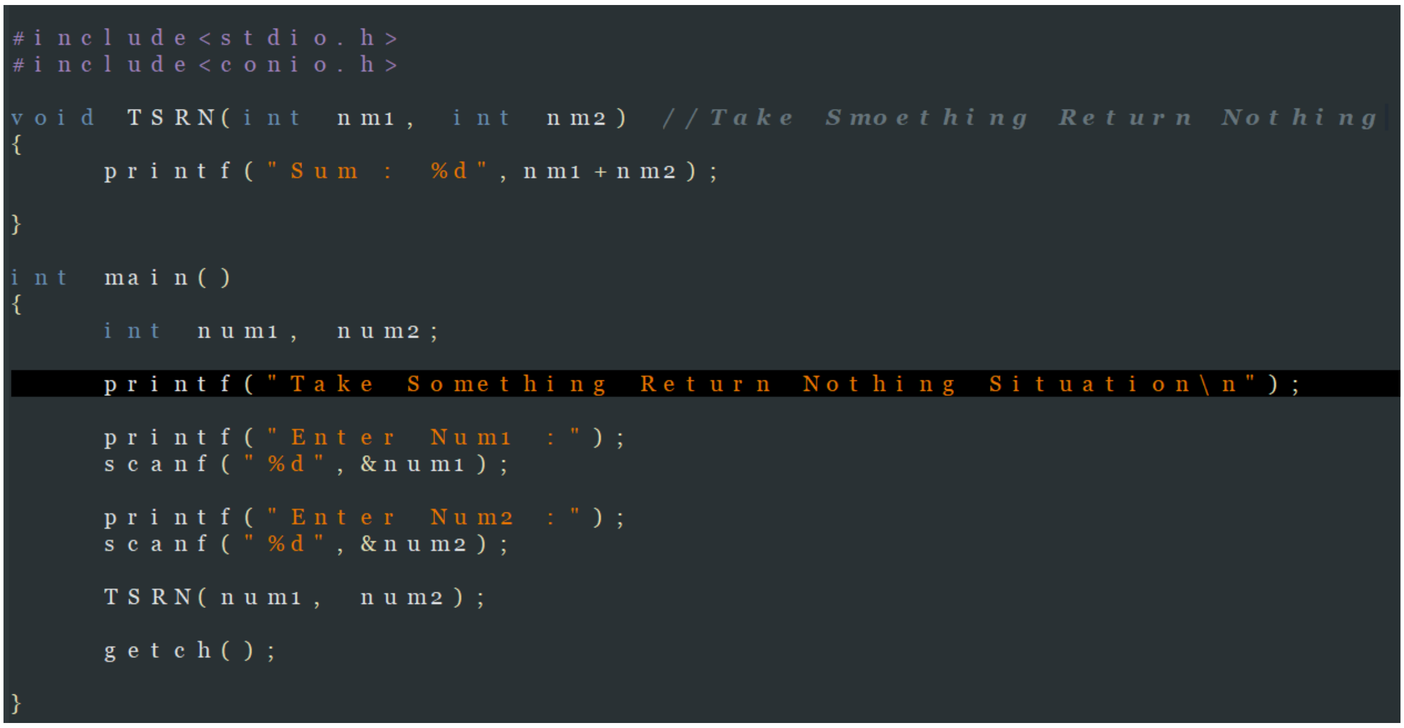
Every C program can contain one or more than one functions. The compiler starts its execution of the program from main() function and generally the execution at the end of the main() function.

Take Nothing Return Nothing Type Function Definition:



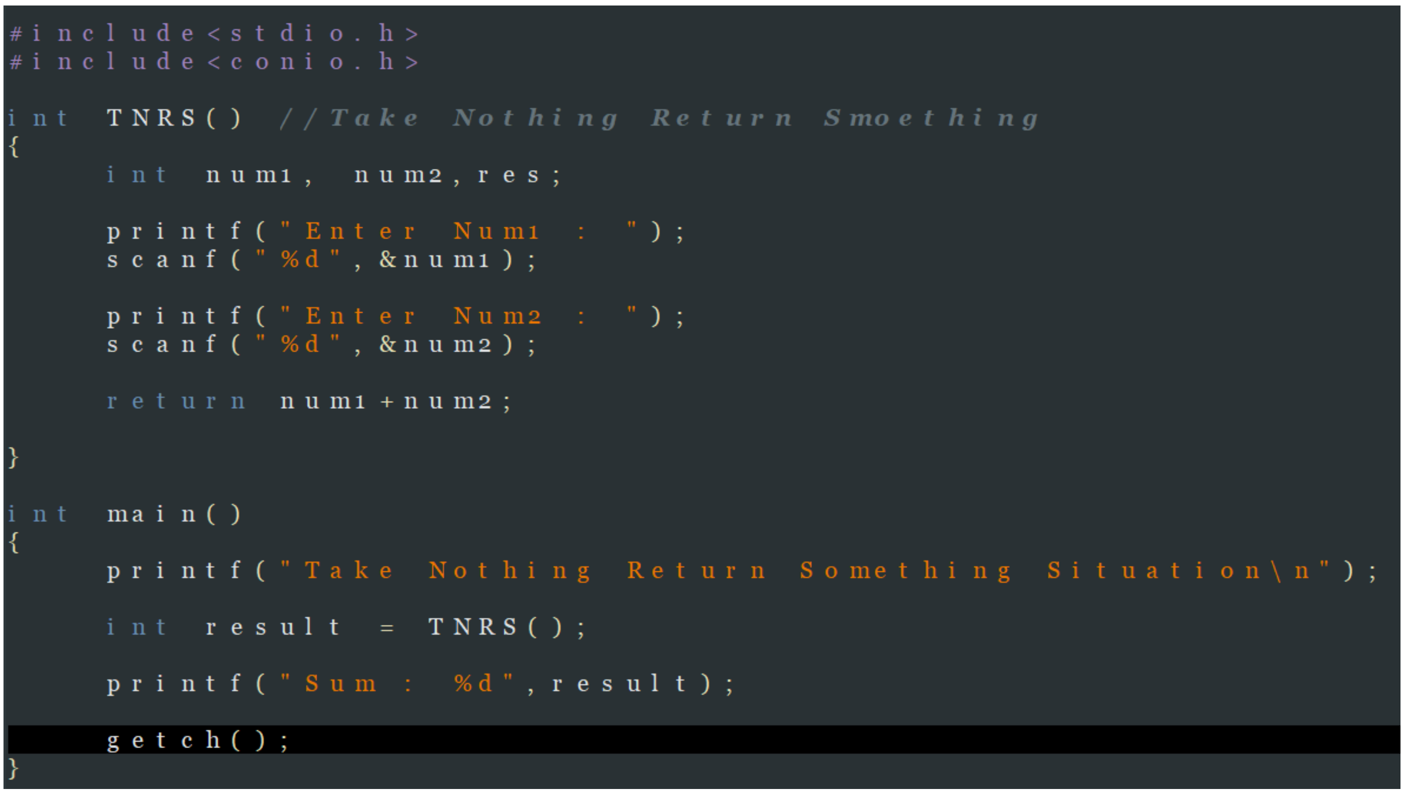
```c
#include<stdio.h>
#include<conio.h>

void TNRN() //Take Nothing Return Nothing
{
	int num1, num2;

	printf("Enter Num1 :  ");
	scanf("%d",&num1);

	printf("Enter Num2 :");
	scanf("%d",&num2);

	printf("Sum : %d",num1+num2);
}

int main()
{
	printf("Take Nothing Return Nothing Situation\n");
	TNRN();

	getch();
}
```

Take Something Return Nothing Type Function Definition:



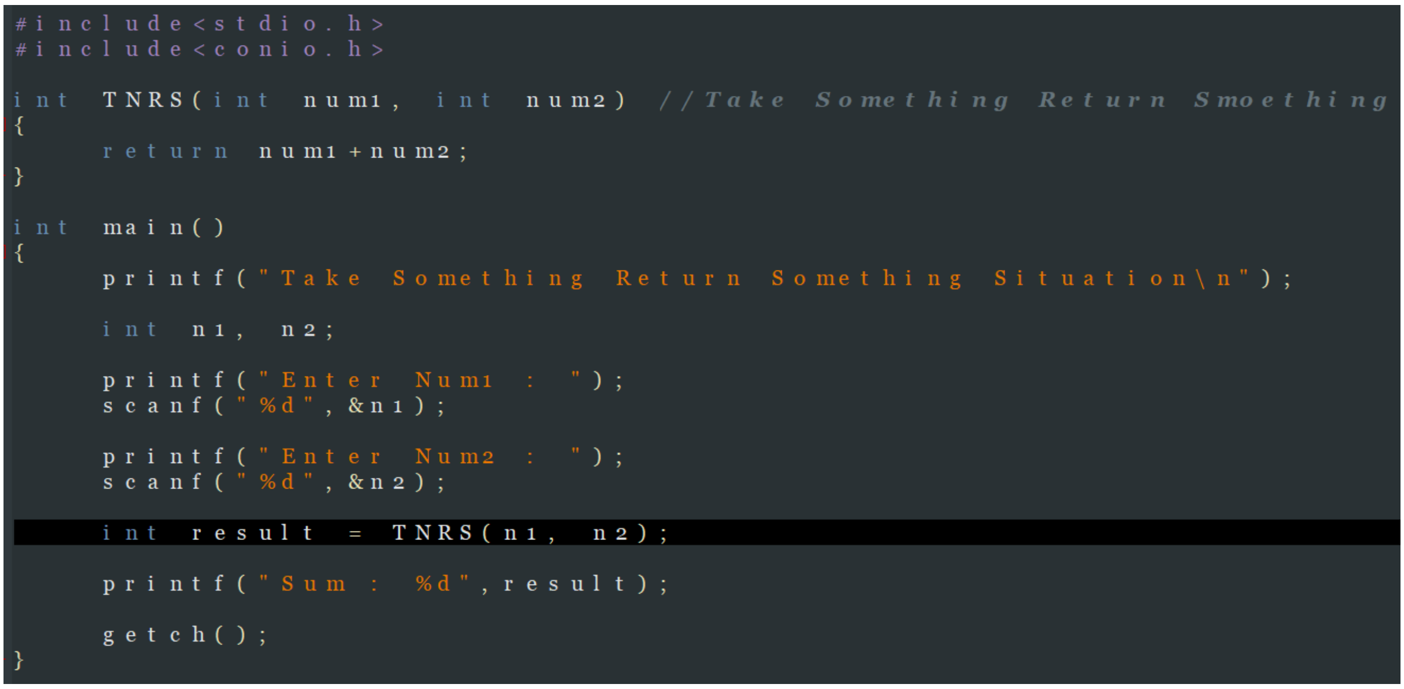
```c
#include<stdio.h>
#include<conio.h>

void TSRN(int nm1, int nm2) //Take Smoething Return Nothing
{
	printf("Sum : %d",nm1+nm2);

}

int main()
{
	int num1, num2;

	printf("Take Something Return Nothing Situation\n");

	printf("Enter Num1 :");
	scanf("%d",&num1);

	printf("Enter Num2 :");
	scanf("%d",&num2);

	TSRN(num1, num2);

	getch();

}
```

Take Nothing Return Something Type Function Definition:



```c
#include<stdio.h>
#include<conio.h>

int TNRS() //Take Nothing Return Smoething
{
    int num1, num2,res;

    printf("Enter Num1 : ");
    scanf("%d",&num1);

    printf("Enter Num2 : ");
    scanf("%d",&num2);

    return num1+num2;

}

int main()
{
    printf("Take Nothing Return Something Situation\n");

    int result = TNRS();

    printf("Sum : %d",result);

    getch();
}
```

Take Something Return Something Type Function Definition:

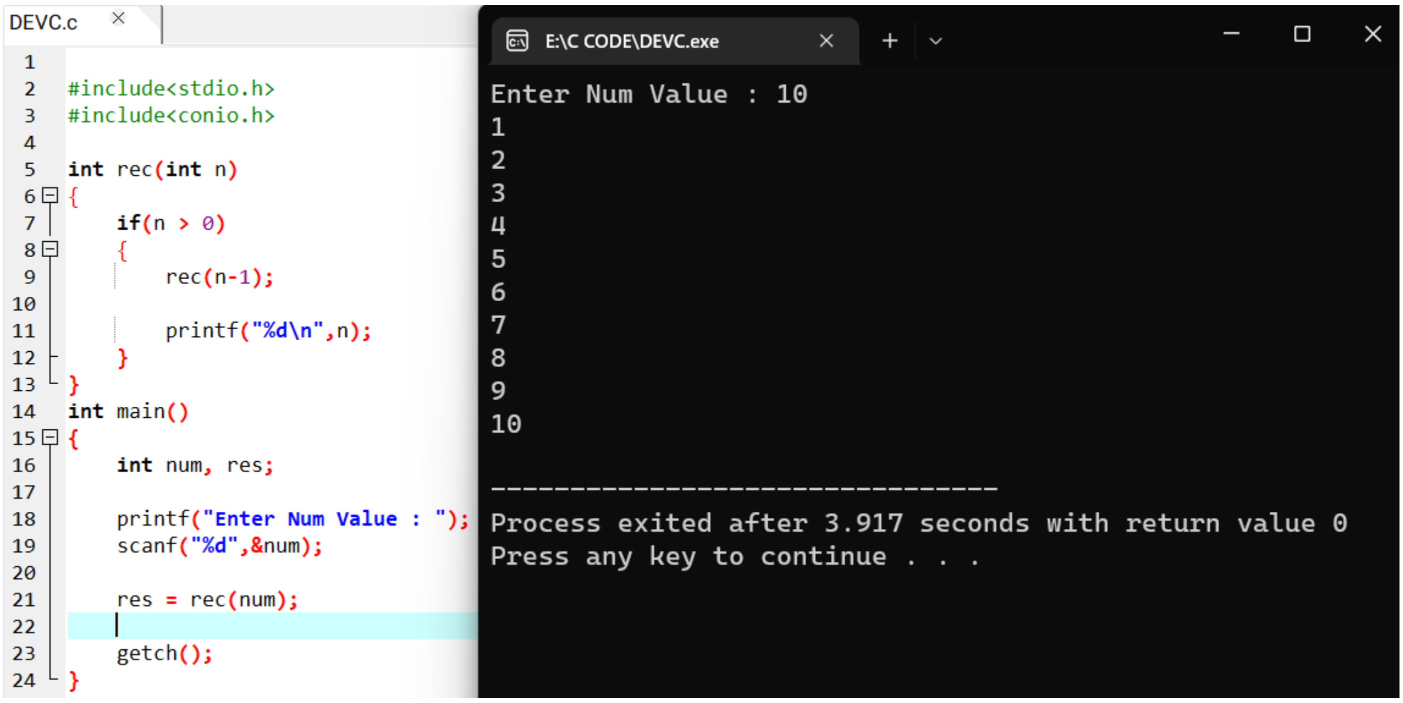


```c
#include<stdio.h>
#include<conio.h>

int TNRS(int num1, int num2) //Take Something Return Smoething
{
    return num1+num2;
}

int main()
{
    printf("Take Something Return Something Situation\n");

    int n1, n2;

    printf("Enter Num1 : ");
    scanf("%d",&n1);

    printf("Enter Num2 : ");
    scanf("%d",&n2);

    int result = TNRS(n1, n2);

    printf("Sum : %d",result);

    getch();
}
```

Recursion In C Programming:

Function calling itself is called recursion. A recursive method solves a problem by calling a copy of itself to work on a smaller problem. It is important to ensure that the recursion terminate. Each time the function call itself with a slightly simpler version of the original problem.

Recursive code is generally shorter and easier to write that iterative code. Solution to some problem are easier to terminate formulate recursively. Recursion is a process in which a function calls itself directly or indirectly.



```c

#include<stdio.h>
#include<conio.h>

int rec(int n)
{
    if(n > 0)
    {
        rec(n-1);

        printf("%d\n",n);
    }
}
int main()
{
    int num, res;

    printf("Enter Num Value : ");
    scanf("%d",&num);

    res = rec(num);

    getch();
}
```

```
Enter Num Value : 10
1
2
3
4
5
6
7
8
9
10

--------------------------------
Process exited after 3.917 seconds with return value 0
Press any key to continue . . .
```