Welcome,

PROGRAMMERS

Types of UDF

TNRN

Take Nothing,
Return Nothing

TNRS

Take Nothing,
Return Something

TSRN

Take Something,
Return Nothing

TSRS

Take Something,
Return Something

Let’s see another remaining two **types of functions** in detail...

03

TNRS

Take Nothing, Return Something

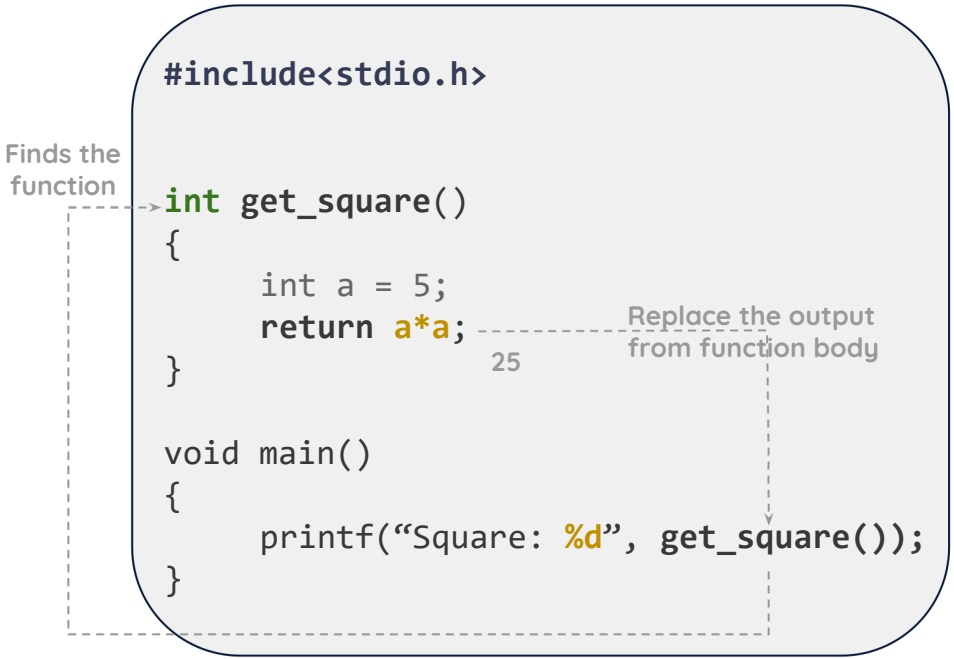
Example of TNRS

```
return_type function_name(parameters)
{
    // Function body (code)
    return value; // Return statement (optional)
}
```

Example

```
int get_square()
{
     int a = 5;
     return a*a;
}
```

Use of TNRS function




```
// Output:
Square: 25
```

04

TSRS

Take Something,
Return Something

Example of TSRS

```
return_type function_name(parameters)
{
    // Function body (code)
    return value; // Return statement (optional)
}
```

Example

```
int get_cube(int a)
{
     return a*a*a;
}
```

Use of TSRN function

```c
#include<stdio.h>

int get_cube(int a)
{
    return a*a*a;
}

void main()
{
    printf("Cube: %d", get_cube(4));
}
```

Passed in Parameters

Replace the output from function body

64

Arguments Passing

```
// Output:
Cube: 64
```

01.

What is Nested Function?

What is Nested Function?

Nested Function

A **nested function** usually refers to **a function that is defined within another function**.
However, it's important to note that true nested functions (functions defined within functions) are **not allowed in C language**.
Each **function** in C **must be independent** and **can't be defined**

inside another function.

Nested Function

Here in C language, kindly note that we can only **call the function within another function**.

We cannot define the function within the function in C language.

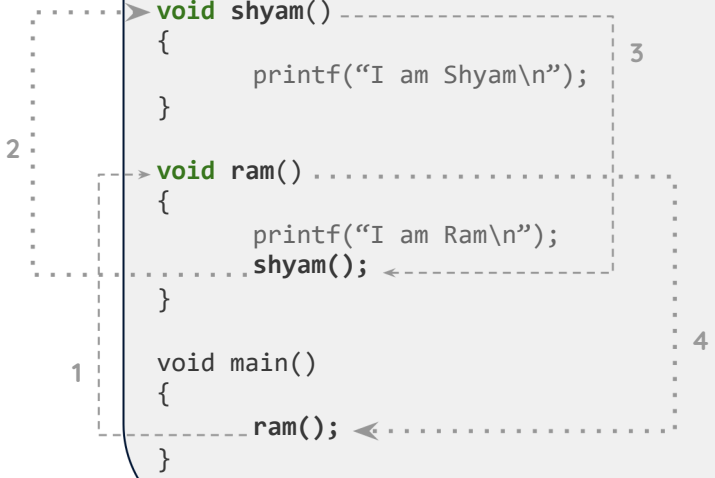
Use of Nested function

```c
#include<stdio.h>

void shyam()
{
        printf("I am Shyam\n");
}

void ram()
{
        printf("I am Ram\n");
        shyam();
}

void main()
{
        ram();
}
```




```
// Output:
I am Ram
I am Shyam
```

02.

What is Recursion?

Recursion

Recursion is a programming concept in which **a function calls itself** directly or indirectly in order to solve a problem.
A function that calls itself is called a **recursive function**, and the **process of repeatedly executing the same set of instructions** is known as **recursion**.

Use of Recursion

```c
#include<stdio.h>

int fact(int n)
{
        if(n <= 1)
                return 1;
        else
                return n*fact(n-1);
}

void main()
{
        printf("Factorial is %d", fact(3));
}
```

```
// Output:
Factorial is 6
```

Language

Let’s start now...