**Type of User-defined Functions in C**

There can be 4 different types of user-defined functions, they are:

1. Function with no arguments and no return value
2. Function with no arguments and a return value
3. Function with arguments and no return value
4. Function with arguments and a return value

**Function with no arguments and no return value**

Such functions can either be used to display information or they are completely dependent on user inputs.

Ex : greatest of two numbers

#include<stdio.h>

void greatNum(); // function declaration

int main()

{

greatNum(); // function call

return 0;

}

**void greatNum()**  // function definition

**{**

int i, j;

printf("Enter 2 numbers that you want to compare...");

scanf("%d%d", &i, &j);

if(i > j) {

printf("The greater number is: %d", i);

}

else {

printf("The greater number is: %d", j);

}

**}**

**Function with no arguments and a return value**

Function returns some result before exiting

Ex ; greatest of two numbers, function will return back value to calling function

#include<stdio.h>

int greatNum(); // function declaration

int main()

{

int result;

**result = greatNum();**  // function call

printf("The greater number is: %d", result);

return 0;

}

int greatNum() // function definition

{

int i, j, greaterNum;

printf("Enter 2 numbers that you want to compare...");

scanf("%d%d", &i, &j);

if(i > j) {

greaterNum = i;

}

else {

greaterNum = j;

}

// returning the result

return greaterNum;

}

**Function with arguments and no return value**

Function which takes input but doesn’t return value back.

Ex : greatest of two numbers

#include<stdio.h>

void greatNum(int a, int b); // function declaration

int main()

{

int i, j;

printf("Enter 2 numbers that you want to compare...");

scanf("%d%d", &i, &j);

greatNum(i, j); // function call

return 0;

}

**void** greatNum(int x, int y) // function definition

{

if(x > y) {

printf("The greater number is: %d", x);

}

else {

printf("The greater number is: %d", y);

}

}

**Function with arguments and a return value**

Function which receives arguments and give back result as well

Ex : greatest of two numbers

#include<stdio.h>

int greatNum(int a, int b); // function declaration

int main()

{

int i, j, result;

printf("Enter 2 numbers that you want to compare...");

scanf("%d%d", &i, &j);

**result** = **greatNum(i, j);** // function call

printf("The greater number is: %d", result);

return 0;

}

**int** greatNum(**int x, int y**) // function definition

{

if(x > y) {

**return** x;

}

else {

**return** y;

}

}

**Recursive Function**

Recursion is a special way of nesting functions, where a function calls itself inside it. We must have certain conditions in the function to break out of the recursion, otherwise recursion will occur infinite times.

**Ex : factorial**

#include<stdio.h>

int factorial(int x); //declaring the function

void main()

{

int a, b;

printf("Enter a number...");

scanf("%d", &a);

b = factorial(a); //calling the function named factorial

printf("%d", b);

}

int factorial(int x) //defining the function

{

int r = 1;

if(x == 1)

return 1;

else

r = x\*factorial(x-1); //recursion, since the function calls itself

return r;

}

**Ex : counter**

#include <stdio.h>

int myFun(int counter)

{

int value = 0;

if(counter == 0){

return;

}else{

value = myFun(counter - 1);

}

printf("%d\n",value);

return counter;

}

int main()

{

myFun(9);

return 0;

}