1) Function Call

```
int italy( )
{
        printf ( "\nI am in italy" ) ;
}

int brazil( )
{
        printf ( "\nI am in brazil" ) ;
}

int argentina( )
{
        printf ( "\nI am in argentina" ) ;
}

int main( )
{
        printf ( "\nI am in main" ) ;
        italy() ;
        brazil() ;
        argentina() ;
}
```

2) One Function being Called by another Functions

```
void italy( )

{

        printf ( "\nI am in italy" ) ;

        brazil( );

        printf ( "\nI am back in italy" ) ;

}



void brazil( )

{

        printf ( "\nI am in brazil" ) ;
```

```
        argentina( ) ;
}

void argentina( )
{
        printf( "\nI am in argentina" ) ;
}

int main( )
{
        printf ( "\nI am in main" ) ;
        italy( );
        printf ( "\nI am finally back in main" ) ;
}
```

3) Main() Function being called by another function

```
#include <stdio.h>
int main();

void hello()
{
        printf("Hello main(), I invite you over Coffee party\n");
        main();
}

int main()
{
        printf("Oh Thanks!\n");
        return 0;
}
```

4) A function can be called any number of times by another functions.

```
#include <stdio.h>

void bscit()
{
        printf("Hello bscitians, am sure all of you are enjoying C today\n");
        printf("Coz today no one is reaching the parallel universe!\n");

}

int main()
{
        printf("Main calls 2 times \n");
        printf("\n");

        bscit(); //1st call
        printf("\n");

        bscit();//2nd call
        return 0;
}
```

5) Order of Function declaration and Order of Function Call does NOT matter

```
#include <stdio.h>

void hey_gals()
{
        printf("Hello gals, ladies first!\n");
}

void hey_boys()
{
        printf("Hello boys, your call is after ladies! \n");

}

void grand_parents()
{
        printf("Hello senior citizens, you can enter at the end, as Boys and gals are faster in
walking\n");
}
int main()
{
```

```
        printf("\n");

        grand_parents(); //1st call
        printf("\n");

        hey_boys(); //2nd call
        printf("\n");

        hey_gals();//3rd call
        return 0;
}
```

6) No Function can be defined another function

```
int main( )
{
        printf ( "\nI am in main" ) ;
        void italy() //intaly() defined in main()
        {
                printf("Hi from Italy\n");
        }
}
```

7) Passing Values between Functions

```
#include <stdio.h>

int sum ( x, y, z ) //receiving the values
int x, y, z ; //optional for the Arguement/parameter datatype
{       //the parameters'/variables' names can be same as in main() also

        int d ;
        d = x + y + z ;
        //return ( d ) ; //without return() also it remains as a fruitful function
}

int subtract(int p, int q, int r) //receiving values with different syntax
{
        int s;
        s = p + q; //Ignored
        s = p + q - r;//Chosen as it exactly matches with number of parameters
        return (s); //fruitful function
}


int main( )
{
```

```
        int x, y, z, final, final1 ;
        //printf ( "\nEnter any three numbers " ) ;
        //scanf ( "%d %d %d", &a, &b, &c ) ;
        final = sum ( 2, 3, 4 ) ; //Value passing for addition
        printf ( "\nSum = %d", final ) ;

        final1 = subtract(2,3,4); //Value passing for subtraction
        printf("\nDifference = %d", final1);
}
```