++

********* C Programming **********

It is a low level programming language, it is developed in 1972 by Dennis Ritchie at the Bell Labs. It is first implemented on UNIX operating system.

What is computer program?
-> It is a set of instruction that is given to the computer to perform any task.

Advantages of C programming language ->

1. It is easy to learn
2. Perform low level activities (Intraction with computer hardware)
3. It can be compiled on various platform or on various computer

==========

Comments in C -> Usually comments are used to explain logic of the code and improve program redability. Comments are ignored by compiler.

Comments can be writen in two ways ->
1. Using // : Single line comments are written after double slash.

Example -
```
#include <stdio.h>
int main()
{

    // Single line Welcome user comment

    printf("Welcome to C programming");
    return 0;
}
```

In above example "// Single line Welcome user comment" is a comment.

2. Using /*.....*/ : Generally Multiline comments are written between slash astrisk and astrisk slash.

Example -
```
#include <stdio.h>
int main()
{

    /* Multiline
       comment
       in
       C programming */


    printf("Welcome to C programming");
```

```
    return 0;
}
```

Assignment Questions

1. Write a Program and Use single line comment in it.
2. Write a program and use multi line comment in it.

==========

printf() in C -> printf() stands for "print formatted". It is used to print any message, value on the screen depending upon the format specifier used.
Format Specifier -> These are used to tell the compiler what type of data is in a variable stored to print or during taking input.

%c -> for character
%d -> for signed integer
%f -> for float
%lf -> for double
%l -> for long

Example -

```
#include<stdio.h>
int main()
{
    int a=50;
    char b='a';
    float c=55.5f;
    double d=72.7;
    printf("%d is an integer\n",a);
    printf("%c is a character\n",b);
    printf("%f is a floating point number\n",c);
    printf("%d is a double\n",d);
    return 0;
}
```

==========

Data types -> Data type specifies which type of value a variable have or to store.

There are following data types -
1. int -> To store integer
2. char -> Stores character
3. float -> To store decimal value
4. double -> It is also used to store real number

Example -

```
#include<stdio.h>

int main()
{
```

```
    int a=5;
    char ch ='a';
    float f = 56.7f;
    double d = 583.3;
    printf("%d \n",a);
    printf("%c \n",ch);
    printf("%f \n",f);
    printf("%lf \n",d);
    return 0;
}
```

==========

Basic Calculation Example

```
#include<stdio.h>
int main()
{
    int a=15;
    int b=7;

    int c=a+b;  // '+' add two numbers

    int d=a-b;  // '-' subtractraction operator

    int e=a*b;  // '*' multiplication operator (astrisk)

    int f=a%b;  // '%' returns the reminder when a is divided by b

    printf("%d  %d  %d  %d ",c,d,e,f);

    return 0;
}
```

Assignment Questions

1. Print sum of two numbers without using 3rd variable
2. Find difference of two numbers
3. print product of two numbers
4. print division of first number by second number
5. Find reminder when 25 is divided by 7
6. Find sum of three numbers
7. Find sum of four numbers
8. Find product of 3 numbers
9. Find product of 4 numbers
10. Find sum of 4 numbers and multiply it by 9 and print
11. Find sum of 4 numbers and divide it by 10 and print
12. find sum of 3 numbers and subtract it from 10000 and print
13. find reminder when 1000 is divided by 222
14. swap two numbers.

Assignment Solutions

1.
```
#include<stdio.h>
int main()
{
    int a=18;
    int b=83;
    printf("%d",a+b);
    return 0;
}
```

2.
```
#include<stdio.h>
int main()
{
    int a=73;
    int b=35;
    printf("%d",a-b);
    return 0;
}
```

3.
```
#include<stdio.h>
int main()
{
    int a=12;
    int b=9;
    printf("%d",a*b);
    return 0;
}
```

4.
```
#include<stdio.h>
int main()
{
    int a=25;
    int b=5;
    printf("%d",a/b);
    return 0;
}
```

5.
```
#include<stdio.h>
int main()
{
    int a=25;
    int b=7;
    printf("%d",a%b);
    return 0;
}
```

==========

Conditional statements -> These are used to perform some tasks only when condition is satisfied.

There are following conditional statements in C :

if -> It executes block of statements only when condition is true.

Example -

```
#include<stdio.h>
int main()
{
    int a = 7;
    if(a<10)
    {
         printf("a is less then 10");
    }
}
```

if else -> It is used when we want to execute block of statement if condition is true and execute any other statements in when condition is false.

Example -

```
#define<stdio.h>
int main()
{
    int num=3;
    if(num>5)
    {
        printf("number is less then 5");
    }
    else
    {
        printf("Number is less then 5");
    }
}
```

if else if ladder -> if-else-if ladder is used to put multiple condition and choose one of them

Example -

```
#define<stdio.h>
int main()
{
    int num=3;
    if(num>5)
    {
        printf("number is greater then 5");
    }
    else if(num==5)
    {
        printf("number is equal to 5");
    }
    else
    {
        printf("Number is less then 5");
    }
    return 0;
}
```

Assignment Questions

1. Find greater number between two numbers.
2. Check weather the number is odd or not.
3. Check weather the number is multiple of 5 or not.
4. Check that the number is divisible by 7.
5. Program to print weekday based on given number. 1 - for monday, 2 - for tuesday, 3 - wednesday, ... 7 - for sunday
6. Check weather the number is positive or negative.
7. Check weather the number is greater then 5000 or not.

1. Find greater between 771 and 717.
2. find smaller between 9192 and 9232
3. check greater then 0 or not.
4. check that the number is greater then 10, less then 10 or equal to 10.
5. check that the number is divisible by 5 or not

Assignment Solutions

1.
```
#include<stdio.h>
int main()
{
    int a=7, b=9;
    if(a<b)
    {
        printf("a is less then b");
    }
    else
    {
        printf("b is less then a");
    }
    return 0;
}
```

2.
```
#include<stdio.h>
int main()
{
    int a=7;
    if(a%2 == 1)
    {
        printf("Number is odd");
    }
    else
    {
        printf("Number is even");
    }
    return 0;
}
```

3.

```
#include<stdio.h>
int main()
{
    int num = 115;
    if(num%5==0)
    {
        printf("Number is multiple of 5");
    }
    else
    {
        printf("Number is not a multiple of 5");
    }
    return 0;
}
```

4.

```
#include<stdio.h>
int main()
{
    int number=147;
    if(number%7 == 0)
    {
        printf("Number is divisible by 7");
    }
    else
    {
        printf("Number is not divisible by 7");
    }
    return 0;
}
```

5.

```
#include<stdio.h>
int main()
{
    int day = 5;
    if(day==1)
    {
        printf("Monday");
    }
    else if(day==2)
    {
        printf("Tuesday");
    }
    else if(day==3)
    {
        printf("Wednesday");
    }
    else if(day==4)
    {
        printf("Thursday");
    }
    else if(day==5)
    {
        printf("Friday");
    }
    else if(day==6)
```

```
    {
        printf("Saturday");
    }
    else if(day==7)
    {
        printf("Sunday");
    }
    else
    {
        printf("Invalid choice");
    }
    return 0;
}
```

6.

```
#include<stdio.h>
int main()
{
    int number=-49;
    if(number >= 0)
    {
        printf("Number is Positive");
    }
    else
    {
        printf("Number is Negative");
    }
    return 0;
}
```

7.

```
#include<stdio.h>
int main()
{
    int num = 5001;
    if(num > 5000)
    {
        printf("Number is greater then 5000");
    }
    else
    {
        printf("Number is less then 5000");
    }
    return 0;
}
```

==========

Ternary Operator -> Ternary operator is shortened way of writing an if-else statement. It takes three arguments.

Example -

```
#include<stdio.h>
int main()
{
    int a=5,b=3;

    a<b ? printf("a is smaller") : printf("b is smaller");
```

```
    return 0;
}
```

Assignment Questions

1. Check positive and negative
2. check even and odd number
3. check divisiblity by 3
4. check equality of two numbers
5. check given character is equal to A or not.

Assignment Solution

1.
```
#include<stdio.h>
int main()
{
    int num = 8;

    num>0 ? printf("Positive"):printf("Negative");

    return 0;
}
```

2.
```
#include<stdio.h>
int main()
{
    int num = 585;

    (num%2==0) ? printf("Even"):printf("Odd");

    return 0;
}
```

3.
```
#include<stdio.h>
int main()
{
    int number = 824;

    (number%3==0) ? printf("Divisible"):printf("Not Divisible");

    return 0;
}
```

4.
```
#include<stdio.h>
int main()
{
    int num1 = 4, num2 = 4;

    (num1==num2) ? printf("Equal"):printf("Not Equal");
```

```
    return 0;
}
```

5.

```
#include<stdio.h>
int main()
{
    char a='A';

    a=='A' ? printf("similar") : printf("different");

    return 0;
}
```

==========

*** && and || and != ***

Logical operators are used to combine two or more conditions.

Example :

```
#include<stdio.h>
int main()
{
  int a=7,b=9;
  int c=5,d=4;
  if(a<b && c>d)
  {
     printf("a is greater then b and c is less then d");
  }
  else if(a==b || c==d)
  {
     printf("Either a is equal to b or c is equal to d");
  }
  else
  {
     printf("Program End");
  }

}
```

Assignment Questions

1. Program to check weather a number is divisible by 2 and 5 both.
2. Program to check weather a number is divisible by 3 and 6 both.
3. Program to find largest number between 3 numbers using logical operators
4. Program to find smallest number between 3 numbers using logical operators
5. Check weather a is equal to b or not using '!=' operator.

Assignment Solutions

1.

```
#include<stdio.h>
int main()
{
    int num = 10;
    if(num%2==0 && num%5==0)
    {
        printf("Number is divisible by 2 and 5 both");
    }
    else
    {
        printf("Number is not divisible by 2 and 5 both");
    }
return 0;
}
```

2.

```
#include<stdio.h>
int main()
{
    int num = 10;
    if(num%3==0 && num%6==0)
    {
        printf("Number is divisible by 2 and 5 both");
    }
    else
    {
        printf("Number is not divisible by 2 and 5 both");
    }
return 0;
}
```

3.

```
#include<stdio.h>
int main()
{
    int num1 = 10,num2=5,num3=4;
    if(num1>=num2 && num1>=num3)
    {
        printf("%d is largest number",num1);
    }
    else if(num2>=num1 && num2>=num3)
    {
        printf("%d is largest number",num2);
    }
    else
    {
        printf("%d is largest number",num3);
    }
return 0;
}
```

4.

```
#include<stdio.h>
int main()
{
    int num1 = 10,num2=5,num3=4;
    if(num1<=num2 && num1<=num3)
    {
        printf("%d is largest number",num1);
```

```
    }
    else if(num2<=num1 && num2<=num3)
    {
        printf("%d is largest number",num2);
    }
    else
    {
        printf("%d is largest number",num3);
    }
return 0;
}
```

5.

```
#include<stdio.h>

int main()
{
  int a=5,b=7;
  if(a!=b)
  {
      printf("a is not equal to b ");
  }
  else
  {
      printf("a is equal to b");
  }
  return 0;
}
```

==========

*** Loops in C ***

Loop is a iterative process which runs continiously till the condition is true.

while loop -> It is used to execute a block of statements till the condition is true.

Example -

```
#include<stdio.h>

int main()
{
  int i=0;
  while(i<10)
  {
      printf("%d",i);
      i++;
  }
  return 0;
}
```

Assignment Questions

1. Print all even numbers from 1 to 30.
2. Print all odd numbers from 1 to 30.
3. Print the table of 5.
4. Find sum of first 10 natural numbers.
5. Find the sum of first 10 even numbers.
6. Find the sum of first 10 odd numbers.
7. Print negative numbers from -10 to -1.
8. Print alphabets from a to z.
9. Print factorial of a number.
10. Print the sum of digit of a number.

1. print numbers 10 to 50
2. print numbers 50 to 10
3. print numbers 1000 to 1100
4. print numbers from 500 to 400
5. print sequence 2,4,6,8....till 50
6. print sequence 50,48,46,44,42...till 2.

Assignment Solutions

1.

```
#include<stdio.h>
int main()
{
  int i=2;
  while(i<=30)
  {
      printf("%d \n",i);
      i=i+2;
  }
  return 0;
}
```

2.

```
#include<stdio.h>
int main()
{
  int i=1;
  while(i<=30)
  {
      printf("%d \n",i);
      i=i+2;
  }
  return 0;
}
```

3.

```
#include<stdio.h>
int main()
{
  int i=5;
  while(i<=50)
  {
      printf("%d \n",i);
      i=i+5;
  }
  return 0;
}
```

4.

```
#include<stdio.h>
int main()
{
  int i=1;
  int sum=0;
  while(i<=10)
  {
      sum=sum+i;
      i++;
  }
  printf("%d",sum);
  return 0;
}
```

5.

```
#include<stdio.h>
int main()
{
  int i=2;
  int sum=0;
  while(i<=20)
  {
      sum=sum+i;
      i=i+2;
  }
  printf("%d",sum);
  return 0;
}
```

6.

```
#include<stdio.h>
int main()
{
  int i=1;
  int sum=0;
  while(i<=20)
  {
      sum=sum+i;
      i=i+2;
  }
  printf("%d",sum);
  return 0;
}
```

7.

```
#include<stdio.h>
int main()
{
  int i=-10;
  while(i<0)
  {
     printf("%d\n",i);
     i=i+1;
  }
  return 0;
}
```

8.

```
#include<stdio.h>
int main()
{
  char i='a';
  while(i<='z')
  {
     printf("%c\n",i);
     i=i+1;
  }
  return 0;
}
```

9.

```
#include<stdio.h>
int main()
{
  int i=5;
  int fact=1;
  while(i>0)
  {
     fact=fact*i;
     i=i-1;
  }
  printf("%d",fact);
  return 0;
}
```

10.

```
#include<stdio.h>
int main()
{
    int num=123;
    int sum=0;
    while(num>0)
    {
       sum=sum + (num%10);
       num=num/10;
    }
    printf("%d",sum);
    return 0;
}
```

==========

do while loop -> It is similar to while, but body of do while is executed atleast once. It is exit controlled loop.

Example -

```
#include <stdio.h>
int main()
{
   int i=0;
   do
   {
      printf("Inside do-while\n");
   }while(i==1);
   return 0;
}
```

Assignment Questions

1. Print Hello World 10 times (use do while loop).
2. Program to print table for the given number using do while loop
3. Print A to Z. (use do while)
4. Print numbers from 20 to 1 in reverse order.(use do while loop)
5. Find sum of digits of any number. (use do while)
6. print square of first five numbers
7. print cube of first 10 numbers

==========

Assignment Solution

1.

```
#include<stdio.h>
int main()
{
    int i=0;
    do{
       printf("Hello World");
       i++;
      }while(i<10);
    return 0;
}
```

2.

```
#include<stdio.h>
int main()
{
    int i=1,number=0;
    printf("Enter a number: ");
    scanf("%d",&number);
```

```
    do{
         printf("%d \n",(number*i));
         i++;
     }while(i<=10);
     return 0;
}
```

3.

```
#include<stdio.h>
int main()
{
    char ch='A';
    do{
       print("%c  ",ch);
       ch++;
      }while(ch<='Z');
    return 0;
}
```

4.

```
#include<stdio.h>
int main()
{
     int num=20;

     do{
          printf("%d  ",num);
          num--;
       }while(num>0);

     return 0;
}
```

5.

```
#include<stdio.h>
int main()
{
     int num=123;
     int sum=0;
     do{
        sum = sum + num%10;
        num = num/10;
       }while(num>0);
     printf("%d", sum);

     return 0;
}
```

6.

```
#include<stdio.h>

int main()
{
    int num=1;
    do
    {
        printf("%d\n",num*num);
        num = num +1;
    }while(num<=5);
```

```
    return 0;
}
```

7.

```
#include<stdio.h>

int main()
{
    int num=1;
    do
    {
        printf("%d\n",num*num*num);
        num = num +1;
    }while(num<=10);
    return 0;
}
```

==========

for loop -> It is used to execute block of statement for known number of times.

Example-

```
#include<stdio.h>
int main()
{
  for(int i=0;i<=10;i++)
  {
      printf("%d  ",i);
  }
  return 0;
}
```

Assignment Questions

1. Print sum of first 10 natural numbers using for loop.
2. Print the sum of first 10 even numbers using for loop.
3. Print the table of entered number.(use for loop)
4. Square of first 10 numbers
5. Cube of first 10 numbers
6. Print number divisible by 2 and 3 both between 1 to 30.(use for loop)
7. Print factorial of a number. (use for loop)
8. Print A to Z
9. Print Z to A

Assignment Solution

1.
```
#include<stdio.h>
int main()
{
    int sum=0;
  for(int i=1;i<=10;i++)
  {
      sum=sum+i;
  }
  printf("%d",sum);
  return 0;
}
```

2.
```
#include<stdio.h>
int main()
{
    int sum=0;
  for(int i=2;i<=20;i=i+2)
  {
      sum=sum+i;
  }
  printf("%d",sum);
  return 0;
}
```

3.
```
#include<stdio.h>
int main()
{
    int num=7;

  for(int i=num;i<=10*num;i=i+num)
  {
      printf("%d  ",i);
  }
  return 0;
}
```

4.
```
#include<stdio.h>

int main()
{
    for(int i=1;i<=10;i++)
    {
        printf("%d\n",i*i);
    }
}
```

5.
```
#include<stdio.h>

int main()
{
    for(int i=1;i<=10;i++)
```

```
    {
        printf("%d\n",i*i*i);
    }
}
```

6.

```
#include<stdio.h>
int main()
{
  for(int i=1;i<=30;i++)
  {
      if(i%2==0 && i%3==0)
      {
          printf("%d ",i);
      }
  }
  return 0;
}
```

7.

```
#include<stdio.h>
int main()
{
     int fact=1;
     for(int i=1;i<=7;i++)
     {
          fact=fact*i;
     }
     printf("%d",fact);
     return 0;
}
```

8.

```
#include<stdio.h>
int main()
{
     for(char i='A';i<='Z';i++)
     {
          printf("%c\t",i);
     }

     return 0;
}
```

9.

```
#include<stdio.h>
int main()
{
     for(char i='Z';i>='A';i--)
     {
          printf("%c\t",i);
     }

     return 0;
}
```

==========

Nested for loop -> Nesting of a loop means using a loop inside another loop

Example -

```
#include<stdio.h>
int main()
{
    for(int i=1;i<5;i++)
    {
        for(int j=1;j<5;j++)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

Assignment Questions

Print Following Patterns -

1.

```
*
**
***
****
*****
```

2.

```
****
***
**
*
```

3.

```
1
2 2
3 3 3
4 4 4 4
```

4.

```
1 1 1 1
2 2 2
3 3
4
```

5.

```
1
1 2
```

```
1 2 3
1 2 3 4

6.
1 2 3 4
1 2 3
1 2
1
```

Assignment Solution

1.
```
#include<stdio.h>
int main()
{
    for(int i=1;i<5;i++)
    {
        for(int j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

2.
```
#include<stdio.h>
int main()
{
    for(int i=5;i>0;i--)
    {
        for(int j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

3.
```
#include<stdio.h>
int main()
{
    for(int i=1;i<5;i++)
    {
        for(int j=1;j<=i;j++)
```

```
        {
            printf("%d ",i);
        }
        printf("\n");
    }

    return 0;
}
```

4.

```
#include<stdio.h>
int main()
{
    for(int i=5;i>0;i--)
    {
        for(int j=1;j<=i;j++)
        {
            printf("%d ",i);
        }
        printf("\n");
    }

    return 0;
}
```

5.

```
#include<stdio.h>
int main()
{
    for(int i=1;i<5;i++)
    {
        for(int j=1;j<=i;j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }

    return 0;
}
```

6.

```
#include<stdio.h>
int main()
{
    for(int i=5;i>0;i--)
    {
        for(int j=1;j<=i;j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }

    return 0;
}
```

==========

Switch -> It is a decision making statement. It is similar to if else if ladder.

Example -

```
#include<stdio.h>
int main()
{
    int a=4;
    switch(a)
    {
        case 1:
        printf("one");
        break;

        case 2:
        printf("Two");
        break;

        case 3:
        printf("Three");
        break;

        case 4:
        printf("Four");
        break;

        case 5:
        printf("Five");
        break;

        default:
        printf("Invalid number");
    }

    return 0;
}
```

Assignment Questions

1. Program to print name of day using switch
   choice 1 for monday
   choice 2 for tuesday....choice 7 for sunday
2. Program to print month
   choice 1 for january
   choice 2 for feburary
   ....choice 12 for december
3. Program to print
   addition of two numbers if choice is 1
   subtraction if choice is 2
   multiplication if choice is 3
   division if choice is 4
4. check weather the alphabet is vowel or consonant using switch case
5. print total number of days in a month using switch case.

6. check weather the number is even or odd using switch
7. Check weather the number is negative or positive.

Assignment Solutions

1.

```
#include<stdio.h>
int main()
{
    int choice=3;
    switch(choice)
    {
        case 1:
        printf("Monday");
        break;

        case 2:
        printf("Tuesday");
        break;

        case 3:
        printf("Wesnesday");
        break;

        case 4:
        printf("Thursday");
        break;

        case 5:
        printf("Friday");
        break;

        case 6:
        printf("Saturday");
        break;

        case 7:
        printf("Sunday");
        break;

        default:
        printf("Invalid number");
    }

    return 0;
}
```

2.

```
#include<stdio.h>
int main()
{
    int choice=3;
    switch(choice)
    {
        case 1:
        printf("January");
        break;
```

```
        case 2:
        printf("February");
        break;

        case 3:
        printf("March");
        break;

        case 4:
        printf("April");
        break;

        case 5:
        printf("May");
        break;

        case 6:
        printf("June");
        break;

        case 7:
        printf("July");
        break;

        case 8:
        printf("August");
        break;

        case 9:
        printf("September");
        break;

        case 10:
        printf("October");
        break;

        case 11:
        printf("November");
        break;

        case 12:
        printf("December");
        break;
        default:
        printf("Invalid number");
    }

    return 0;
}
```

3.

```
#include<stdio.h>
int main()
{
    int a=9,b=3;
    int choice=3;
    switch(choice)
```

```
    {
        case 1:
        printf("%d ",a+b);
        break;

        case 2:
        printf("%d ",a-b);
        break;

        case 3:
        printf("%d ",a*b);
        break;

        case 4:
        printf("%d ",a/b);
        break;

        default:
        printf("Invalid number");
    }

    return 0;
}
```

4.

```
#include<stdio.h>

int main()
{
    char ch='g';
    switch(ch)
    {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U':
        printf("It is a vowel");
        break;
        default:
        printf("Not a vowel");
    }

}
```

5.

```
#include<stdio.h>

int main()
{
    int choice = 7;
    switch(choice)
    {
```

```
        case 1:
        printf("31 days");
        break;
        case 2:
        printf("28/29 days");
        break;
        case 3:
        printf("31 days");
        break;
        case 4:
        printf("30 days");
        break;
        case 5:
        printf("31 days");
        break;
        case 6:
        printf("30 days");
        break;
        case 7:
        printf("31 days");
        break;
        case 8:
        printf("31 days");
        break;
        case 9:
        printf("30 days");
        break;
        case 10:
        printf("31 days");
        break;
        case 11:
        printf("30 days");
        break;
        case 12:
        printf("31 days");
        break;

        default:
        printf("Invalid choice");
    }

}

6.
#include<stdio.h>

int main()
{
    int num = 7;
    switch(num%2)
    {
        case 1:
        printf("Number is odd");
        break;
        case 0:
        printf("Number is even");
        break;
    }
}
```

7.
```
#include<stdio.h>

int main()
{
    int num = -12;
    int choice;
    if(num<0)
    {
        choice = 1;
    }
    else
    {
        choice = 0;
    }
    switch(choice)
    {
        case 0:
        printf("Number is Positive");
        break;
        case 1:
        printf("Number is Negative");
        break;
    }
}
```

==========

Function in C -> Function is a collection of statements which are grouped together to perform any task.

Example -

```
#include<stdio.h>

void func()              //Function definition
{
   printf("I am a function");
}

int main()
{
    func();      //Function calling
}
```

Why we use function?
1. Function are generally used for reusability of a code.
2. To reduce complexity of a program.
3. To increase the redability of a code.

void function -> Function which does not return any value are called void function.
These are defined using void keyword.

Example -

```
#include<stdio.h>

void function()
{
   printf("I am a function");
}

int main()
{
    function();
    return 0;
}
```

Function with return value -> These function return any value like integer,float, etc.

Example -

```
#include<stdio.h>

int function()
{
   return 7;
}

int main()
{
    printf("%d",function());
    return 0;
}
```

Assignment Questions

1. Write a function to calculate the addition of two numbers.
2. Write a function to return area of a rectangle.
3. Write a function to print if a number is even or odd.
4. Write a function to return perimeter of a rectangle.
5. Write a function to return a factorial of given number.
6. Write a function to print weather the number is posititve or negative.
7. function to check he/she is elegible to vote.
8. function to take 3 parameters and find average of that numbers.

Assignment Solutions

1.

```
#include<stdio.h>

int function(int a,int b)
```

```
{
   return a+b;
}

int main()
{
    printf("%d",function(5,3));
    return 0;
}
```

2.

```
#include<stdio.h>

int area(int len,int bre)
{
   return len*bre;
}

int main()
{
    printf("%d",area(5,3));
    return 0;
}
```

3.

```
#include<stdio.h>

void even_odd(int num)
{
   if(num%2==0)
   printf("Number is even");
   else
   printf("Number is odd");
}

int main()
{
    even_odd(573);
    return 0;
}
```

4.

```
#include<stdio.h>

int perimeter(int l,int b)
{
   return 2*(l+b);
}

int main()
{
    printf("%d",perimeter(5,13));
    return 0;
}
```

5.

```
#include<stdio.h>

int factorial(int num)
```

```
{
    int fact=1;
    for(int i=1;i<=num;i++)
    {
        fact=fact*i;
    }
    return fact;
}

int main()
{
    printf("%d",factorial(5));
    return 0;
}
```

6.

```
#include<stdio.h>

int function(int num)
{
    if(num>=0)
    printf("Number is positive");
    else
    printf("Number is negative");
}

int main()
{
    function(-544);
    return 0;
}
```

7.

```
#include<stdio.h>

int function(int age)
{
    if(age<18)
    printf("Not elegible to vote");
    else
    printf("Elegible to vote");
}

int main()
{
    function(13);
    return 0;
}
```

8.

```
#include<stdio.h>

float average(int x,int y,int z)
{
    int avg;
    avg=(x+y+z)/3;
    return avg;
}
```

```
int main()
{
    printf("%f",average(6,22,29));
    return 0;
}
```

==========

Macros in C -> They are used when very small code or any value is used repeatedly many times. They are defined by "#define directive".

Macro are replaced by their value or code before compilation. These are used to inhance performance of program.

Example -

```
#include<stdio.h>
#define PI 3.14
int main()
{
   float radius=6;
   radius =2*PI*radius;
   printf("%f",radius);
}
```

```
#include<stdio.h>
#define Area(l,b) (l*b)

int main()
{
   int area;
   area = Area(5,3);
   printf("%d", area);
   return 0;
}
```

Assignment Questions

1. Define a macro with value 21.77 and print it.
2. Define two macro in same program and print them in different line.
3. Create a macro to find perimeter of circle.
4. Create a macro to find area of circle.
5. Create a macro to find perimeter of rectangle.
6. Macro to find average of two numbers.
7. Macro to find square of a number.

Assignment Solutions

1.
```
#include<stdio.h>
#define VALUE 21.77
int main()
{
    printf("%f",VALUE);
    return 0;
}
```

2.
```
#include<stdio.h>
#define M1 824
#define M2 6.122
int main()
{
    printf("%d \n%f",M1,M2);
    return 0;
}
```

3.
```
#include<stdio.h>
#define PI 3.14
#define PERIMETER(r) (2*PI*r)
int main()
{
    float peri;
    peri = PERIMETER(7);
    printf("%f",peri);
    return 0;
}
```

4.
```
#include<stdio.h>
#define PI 3.14
#define AREA(r) (PI*r*r)
int main()
{
    float area;
    area = AREA(7);
    printf("%f",area);
    return 0;
}
```

5.
```
#include<stdio.h>

#define PERIMETER(l,b) (2*(l+b))
int main()
{
    int peri;
    peri = PERIMETER(11,7);
    printf("%d",peri);
    return 0;
}
```

6.
```
#include<stdio.h>
```

```
#define Avg(a,b) ((a+b)/2)
int main()
{
    float average;
    average = Avg(15,17);
    printf("%f",average);
    return 0;
}
```

7.

```
#include<stdio.h>

#define square(num) (num*num)
int main()
{
    printf("%d",square(15));
    return 0;
}
```

==========

Pointers in C -> Pointer are the datatype which stores the address of any operator. These are declared using *(astrisk) after data type of variable.

Example -

1.

```
#include<stdio.h>
int main()
{
    int var=7;
    int* ptr;    //Pointer decleration
    ptr=&var;    //Pointer assignment
    printf("%x\n",ptr);
    return 0;
}
```

2.

```
#include<stdio.h>
int func(int* a,int* b)
{
    return (*a + *b);
}
int main()
{
    int a=7,b=21;
    int c=func(&a,&b);
    printf("%d",c);
    return 0;
}
```

Assignment Questions

1. Print address and value of double type variable.

2. Write a program to subtraction of two numbers using pointers.
3. Write a program to multiply two numbers using pointers.
4. Find the maximum number between two numbers using a pointer.
5. Function to return average of three numbers (use pointer).
6. Swap two numbers using pointer.
7. Division of one number by another number.(use pointer)

Assignment Solutions

1.
```
#include<stdio.h>

int main()
{
    double num = 47.34;
    double* ptr;
    ptr = &num;
    printf("%x \n",ptr);
    printf("%lf",*ptr);
    return 0;
}
```

2.
```
#include<stdio.h>

int main()
{
    int a=53,b=23;
    int* ptr1 = &a,*ptr2 = &b;
    int c;
    c=*ptr1 - *ptr2;
    printf("%d",c);
    return 0;
}
```

3.
```
#include<stdio.h>

int main()
{
    int a=5,b=20;
    int* ptr1 = &a,*ptr2 = &b;
    int c;
    c=*ptr1 * *ptr2;
    printf("%d",c);
    return 0;
}
```

4.
```
#include<stdio.h>
```

```
int main()
{
    int a=5,b=20;
    int* ptr1 = &a,*ptr2 = &b;
    if(*ptr1>*ptr2)
    {
        printf("%d",*ptr1);
    }
    else
    {
        printf("%d",*ptr2);
    }
    return 0;
}
```

5.

```
#include<stdio.h>

float average(int* x,int* y,int* z)
{
    return (*x+*y+*z)/3;
}
int main()
{
    int a=5,b=20,c=11;
    float avg;
    avg = average(&a,&b,&c);
    printf("%f",avg);

    return 0;
}
```

6.

```
#include<stdio.h>

int main()
{
    int a=5,b=20;
    int* ptr1,*ptr2;
    ptr1=&a;
    ptr2=&b;
    printf("Before Swaping %d  %d\n",*ptr1,*ptr2);
    int* temp;
    temp = ptr1;
    ptr1 = ptr2;
    ptr2 = temp;
    printf("After Swaping %d  %d",*ptr1,*ptr2);

    return 0;
}
```

7.

```
#include<stdio.h>

int main()
{
    int num1 = 21,num2=7;
    int* ptr1=&num1;
```

```
    int* ptr2=&num2;
    int ans;
    ans = *ptr1/(*ptr2);
    printf("%d",ans);
    return 0;
}
```

==========

Arrays in C -> Array is a collection of same type of data which are stored in continious memory location.
Array is derived data type, means it is derived from built-in (primitive) data types like int,float,char,etc.

Array decleration syntax -> DataType Array_Name[Size_of_array];

Example -
1.

```
#include<stdio.h>

int main()
{
    int arr[5]={1,2,3,4,5};
    printf("%d\n",arr[0]);
    printf("%d\n",arr[1]);
    printf("%d\n",arr[2]);
    printf("%d\n",arr[3]);
    printf("%d\n",arr[4]);
    return 0;
}
```

2.

```
#include<stdio.h>

int main()
{
    int arr[3][3]={1,2,3,4,5,6,7,8,9};
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            printf("%d ",arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Assignment Questions

1. Create array of 5 character and print it.
2. Create array of 10 integers and find sum of all elements.
3. Array of 7 element and product of all.
4. Array of first 5 even number and print it.
5. Array of first 5 odd number and print sum.
6. Store table of 4 in array and print it.
7. Create 3*3 matrix with all odd values and print it.

Assignment Solutions

1.
```
#include<stdio.h>

int main()
{
    char arr[5]={'a','b','c','d','e'};
    for(int i=0;i<5;i++)
    {
        printf("%c  ",arr[i]);
    }
    return 0;
}
```

2.
```
#include<stdio.h>

int main()
{
    int arr[10]={1,2,3,2,5,1,6,9,4,7};
    int sum=0;
    for(int i=0;i<10;i++)
    {
        sum=sum+arr[i];
    }
    printf("%d",sum);
    return 0;
}
```

3.
```
#include<stdio.h>

int main()
{
    int arr[7]={1,2,2,6,9,4,7};
    int product=1;
    for(int i=0;i<7;i++)
    {
        product=product+arr[i];
    }
    printf("%d",product);
    return 0;
}
```

4.
```
#include<stdio.h>

int main()
{
    int arr[5]={2,4,6,8,10};
    for(int i=0;i<5;i++)
    {
         printf("%d  ",arr[i]);
    }
    return 0;
}
```

5.
```
#include<stdio.h>

int main()
{
    int arr[5]={1,3,5,7,9};
    int sum=0;
    for(int i=0;i<5;i++)
    {
         sum=sum+arr[i];
    }
    printf("%d",sum);
    return 0;
}
```

6.
```
#include<stdio.h>

int main()
{
    int table[10];
    int cnt=0;
    for(int i=4;i<=40;i=i+4)
    {
         table[cnt]=i;
         cnt++;
    }
    for(int i=0;i<10;i++)
    {
        printf("%d\n",table[i]);
    }
    return 0;
}
```

7.
```
#include<stdio.h>
int main()
{
    int matrix[3][3]={1,3,5,7,9,11,13,15,17};
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            printf("%d  ",matrix[i][j]);
        }
        printf("\n");
```

```
    }
    return 0;
}
```

==========

String in C -> String is the collection of characters, which is terminated by null
character(\0).
Difference between character array and string is that string is
terminated by null character where char array is not.

Example -
1.
```
#include<stdio.h>

int main()
{
    char str[10] = "number";   // string decleration and intialization
    printf("%s",str);
    return 0;
}
```

2.
```
#include<stdio.h>

int main()
{
    char str[] = "Welcome to C programming";
    printf("%s",str);
    return 0;
}
```

3.
```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[]="Hello";
    char str2[]="World";
    char str3[]="";
    printf("%d \n",strlen(str1));

    //strcpy(str3,str1);
    //printf("%s \n",str3);

    //strcat(str1,str2);
    //printf("%s \n",str1);

    //int a = strcmp(str1,str2);
    //printf("%d",a);
    return 0;
}
```

Assignment Questions

1. Create a string and print it on the screen.
2. Program to find length of string without using strlen().
3. Program to seperate each cahracter from string.
4. count total words in a string and print it.
5. Reverse a string.
6. Take two string and copy the content of 1st into second without strcpy().
7. Print occurence of E in string "WELCOME TO THE C PROGRAMMING WORLD".

Assignment Solutions

2.

```
#include<stdio.h>

int main()
{
    char str[] = "Welcome to C programming";
    int count=0;
    for(int i=0;str[i]!='\0';i++)
    {
        count=count + 1;
    }
    printf("%d",count);
    return 0;
}
```

3.

```
#include<stdio.h>

int main()
{
    char str[] = "programming";
    for(int i=0;str[i]!='\0';i++)
    {
        printf("%c  ",str[i]);
    }

    return 0;
}
```

4.

```
#include<stdio.h>

int main()
{
    char str[] = "Total number of words in the string";
    int word=0;
    for(int i=0;str[i]!='\0';i++)
    {
        if(str[i]==' ')
```

```
        word++;
    }
    word++;
    printf("%d",word);
    return 0;
}
```

5.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[] = "documentation";
    char rev[]="";
    int len;
    len = strlen(str);
    for(int i=len-1;i>=0;i--)
    {
        printf("%c",str[i]);
    }
    printf("%s",rev);
    return 0;
}
```

6.

```
#include<stdio.h>

int main()
{
    char str1[] = "documentation";
    char str2[] = "";
    int i;
    for(i=0;str1[i]!='\0';i++)
    {
        str2[i]=str1[i];
    }
    str2[i]='\0';
    printf("%s",str2);
    return 0;
}
```

7.

```
#include<stdio.h>

int main()
{
    char str[] = "WELCOME TO THE C PROGRAMMING WORLD";
    int cnt=0;
    for(int i=0;str[i]!='\0';i++)
    {
        if(str[i]=='E')
        cnt++;
    }
    printf("%d",cnt);
    return 0;
}
```

==========

Structure in C -> Structures are the user defined data types in C.
These are used to group different type of information under same name.

How to define structure?
-> Structure are defined by using struct keyword.

Syntax ->

```
struct structure_name
{
    // structure members
};
```

// Structure members are accessed by using structure member operator also called dot
operator(.)

Example -

1. First method of assigning.

```
#include<stdio.h>
struct Student
{
    int RollNo;
    int Marks;
};

int main()
{
    struct Student Rohit,Sunil,Anshul;
    Rohit.RollNo = 1;
    Sunil.RollNo = 2;
    Anshul.RollNo = 3;
    Rohit.Marks = 85;
    Anshul.Marks = 77;
    Sunil.Marks = 57;
    printf("RollNo %d gets %d Marks\n",Rohit.RollNo,Rohit.Marks);
    printf("RollNo %d gets %d Marks\n",Sunil.RollNo,Sunil.Marks);
    printf("RollNo %d gets %d Marks",Anshul.RollNo,Anshul.Marks);
    return 0;
}
```

2. Another method of assigning values to data members.

```
#include<stdio.h>
struct Student
{
    char name[20];
    int RollNo;
};

int main()
{
    struct Student Rohit={"Rohit",1},Sunil={"Sunil",2},Anshul={"Anshul",3};
```

```
    printf("RollNo %d gets %d Marks\n",Rohit.name,Rohit.Marks);
    printf("RollNo %d gets %d Marks\n",Sunil.name,Sunil.Marks);
    printf("RollNo %d gets %d Marks",Anshul.name,Anshul.Marks);
    return 0;
}
```

3. structure pointer

```
#include<stdio.h>
struct Student
{
    int RollNo;
    int Marks;
};

int main()
{
    struct Student s1,s2;
    struct Student* ptr1,*ptr2;
    ptr1 = &s1;
    ptr2 = &s2;
    ptr1->RollNo = 1;
    ptr2->RollNo = 2;
    ptr1->Marks = 56;
    ptr2->Marks = 96;
    printf("%d  %d\n",ptr1->RollNo,ptr1->Marks);
    printf("%d  %d\n",ptr2->RollNo,ptr2->Marks);
    return 0;
}
```

4. Structure array

```
#include<stdio.h>
struct Student
{
    int RollNo;
    int Marks;
};

int main()
{
    struct Student s[5]={1,73,2,82,3,52,4,90,5,63};
    printf("%d   %d \n",s[0].RollNo,s[0].Marks);
    printf("%d   %d \n",s[1].RollNo,s[1].Marks);
    printf("%d   %d \n",s[2].RollNo,s[2].Marks);
    printf("%d   %d \n",s[3].RollNo,s[3].Marks);
    printf("%d   %d \n",s[4].RollNo,s[4].Marks);
    return 0;
}
```

Assignment Questions

1. create Structure student contain Name,rollno,marks of 3 students and display it.
2. create structure book contain book title, bookid,price of 3 books and display it.
3. Structure employee contain name, salary of 4 employee and print it.
4. Structure grocery contain name and price of 3 item and print it.
5. print name,rollno,marks of 3 student using structure pointer.
6. create structure student using structure array contain Name,rollno,marks of 7 students and display it

Assignment Solutions

1.

```
#include<stdio.h>
#include<string.h>
struct Student
{
    char name[20];
    int RollNo;
    int Marks;
};

int main()
{
    struct Student s1,s2,s3;
    strcpy(s1.name,"Amit");
    strcpy(s2.name,"Gourav");
    strcpy(s3.name,"Sumit");
    s1.RollNo = 1;
    s2.RollNo = 2;
    s3.RollNo = 3;
    s1.Marks = 86;
    s2.Marks = 96;
    s3.Marks = 43;
    printf("%s  %d  %d\n",s1.name,s1.RollNo,s1.Marks);
    printf("%s  %d  %d\n",s2.name,s2.RollNo,s2.Marks);
    printf("%s  %d  %d\n",s3.name,s3.RollNo,s3.Marks);
    return 0;
}
```

2.

```
#include<stdio.h>
#include<string.h>
struct Book
{
    char BookName[20];
    char bookid[20];
    int price;
};
```

```
int main()
{
    struct Book B1={"Harry Potter","abc123",200};
    struct Book B2={"Wings of fire","1ab2c3",120};
    struct Book B3={"Believe in Yourself","aaa77",160};
    printf("%s   %s   %d\n",B1.BookName,B1.bookid,B1.price);
    printf("%s   %s   %d\n",B2.BookName,B2.bookid,B2.price);
    printf("%s   %s   %d\n",B3.BookName,B3.bookid,B3.price);

    return 0;
}
```

3.

```
#include<stdio.h>
#include<string.h>
struct Employee
{
    char name[20];
    int salary;
};

int main()
{
    struct Employee E1={"Harry",20000};
    struct Employee E2={"John",15000};
    struct Employee E3={"Peter",45000};
    struct Employee E4={"James",30000};
    printf("%s     %d\n",E1.name,E1.salary);
    printf("%s      %d\n",E2.name,E2.salary);
    printf("%s     %d\n",E3.name,E3.salary);
    printf("%s     %d ",E4.name,E4.salary);

    return 0;
}
```

4.

```
#include<stdio.h>
#include<string.h>
struct Grocery
{
    char name[20];
    int price;
};

int main()
{
    struct Grocery G1={"Wheat",500};
    struct Grocery G2={"Rice",400};
    struct Grocery G3={"Oil",750};

    printf("%s     %d\n",G1.name,G1.price);
    printf("%s      %d\n",G2.name,G2.price);
    printf("%s     %d\n",G3.name,G3.price);


    return 0;
}
```

5.

```
#include<stdio.h>
#include<string.h>
struct Student
{
    char name[20];
    int RollNo;
    int Marks;
};

int main()
{
    struct Student s1,s2,s3;
    struct Student* ptr1,*ptr2,*ptr3;
    ptr1 = &s1;
    ptr2 = &s2;
    ptr3 = &s3;
    strcpy(ptr1->name,"Amit");
    strcpy(ptr2->name,"Gourav");
    strcpy(ptr3->name,"Sumit");
    ptr1->RollNo = 1;
    ptr2->RollNo = 2;
    ptr3->RollNo = 3;
    ptr1->Marks = 86;
    ptr2->Marks = 96;
    ptr3->Marks = 43;
    printf("%s  %d  %d\n",ptr1->name,ptr1->RollNo,ptr1->Marks);
    printf("%s  %d  %d\n",ptr2->name,ptr2->RollNo,ptr2->Marks);
    printf("%s  %d  %d\n",ptr3->name,ptr3->RollNo,ptr3->Marks);
    return 0;
}
```

6.

```
#include<stdio.h>
#include<string.h>
struct Student
{
    char name[20];
    int rollno;
    int marks;
};

int main()
{
    struct Student s[7];
    strcpy(s[0].name,"Ashu");
    strcpy(s[1].name,"Ayush");
    strcpy(s[2].name,"Aman");
    strcpy(s[3].name,"Deepak");
    strcpy(s[4].name,"Ishan");
    strcpy(s[5].name,"Kunal");
    strcpy(s[6].name,"Mayank");
    s[0].rollno = 1;
    s[1].rollno = 2;
    s[2].rollno = 3;
    s[3].rollno = 4;
    s[4].rollno = 5;
    s[5].rollno = 6;
    s[6].rollno = 7;
    s[0].marks = 65;
```

```
    s[1].marks = 73;
    s[2].marks = 58;
    s[3].marks = 92;
    s[4].marks = 84;
    s[5].marks = 70;
    s[6].marks = 77;
    for(int i=0;i<7;i++)
    {
        printf("%s  %d   %d\n",s[i].name,s[i].rollno,s[i].marks);
    }
    return 0;
}
```

==========

File Handling -> It is a method of working with the files using a program.

Why to use file?
-> 1. When a program is terminated then the entire data is lost, to prevent this
      we use file to save data even after the program is terminated.
   2. With the help of file we can transfer data from one computer to another.

In C, you can perform four major operations on files:

-> Creating a new file
-> Opening an existing file
-> Closing a file
-> Reading and writing information to a file

File can be opened using fopen() function.
 fopen("file","mode");

mode -> w - for writing
        r - for reading
        a - for appending