Compilation & Execution of C Program :

* Compiling is the transformation from Source Code i.e. human readable code into machine code i.e. Computer executable.

*. A compiler is a program.

A compiler takes the Source code written in High Level language and transforms this code into machine language that can be understoop by Computer itself.

* The compiler ensures that your program is TYPE correct.
* The compiler ensures that your program is syntactically correct.
* The compiler ensures that your program is logically correct.

Steps for Compilation & Execution

Step1. ① Write the source code in any Text Editor (like notepad)

Step2 : Save the file as filename.c extension.
File > Save as > A dialog box will appear > Choose the desired Location where you want to save the source code file > Click on Save button.

Step3 : Open Command Prompt.

Start > Run > A Run panel will open

type cmd

press enter.

Step4 : Locate your pointer to the location where you saved your source code of C file.

Cd "path/"

cd - change directory

Ex: `cd ..` // to locate one folder backword

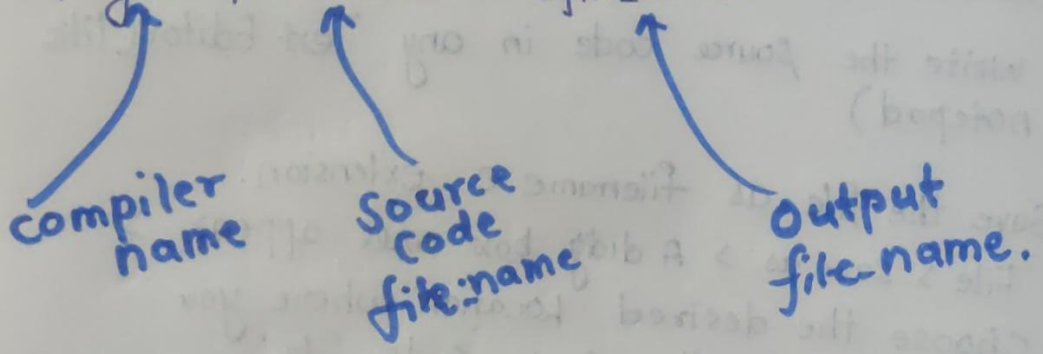
cd "C:\\ users \\ Desktop"

Step-5

after locating to desired position (folder | Location | directory)

type

> gcc filename.C -ofile_name



press enter.

(your source code will be compiled).

Step 6:

To execute the compiled code type in cmd

> file_name

press enter

* The desired output will be prompted to the output screen (cmd).

Prg 1#

W.A.P. in C, to check whether a given number is even or odd.

```
#include<stdio.h>
void main()
{
  int num;
  printf("Enter a number\n");
  Scanf("%d", &num);
if(num%2==0)
   {
printf("The given number %d is even", num);
   }
else
   {
printf("The given number %d is odd", num);
   }
   getch()
}.
```

Prog. #2

W.A.P. in C to print Sum of first n numbers

```
#include<Stdio.h>
void main( )
{
  int  n, sum=0;
  int i;
  printf(" Enter the value of n /n ");
  scanf ("%d", &n);
  for( i=1 ; i<=n; i++)
     {
       sum= sum+n;
     }
  printf ( "The sum of first %d number is %d",
          n , sum );
  getch( );
}
```

Prg 3#.

W.A.P. in C to print sum of digits of a number.

```
#include <stdio.h>
#include <conio.h>
void main( )
{
  int num, sum=0, rem;
printf ("Enter a number \n");
scanf ("%d", &num);
while ( num != 0 )
   {
     rem = num % 10;
    sum = sum + rem;
    num = num / 10;
   }
printf ("sum of digits of entered number
 = %d \n", sum);
getch( )
}
```

Prg: 4:

W.A.P. in C to reverse the digits of a number.

```
#include <stdio.h>
#include <conio.h>
void main()
  {
    int num, r
   printf("Enter a number")
   printf("\n");
   scanf("%d", &num);
   printf("Reverse of the number is\n");

do{
  r=num%10;
   printf("%d", r);
   num=num/10;
 }while(num>0);
 getch()
}
```

Prg. 4.

W.A.P. in C to print fibonacci numbers.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 141...

$f_0 = 0$

$f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$

```
#include < stdio.h >
# include < conio.h >
void main( )
{ int i;
  int n;
  int a = 0;
  int b = 1;
  int c;
 printf ( "Enter the limit ");
 scanf ( "%d ", &n);
  for( i=0; i<=n; i++)
    {
      printf( "%d \n", a);
        c = a+3;
        a = b;
        b = c;
```

```
}
  getch();
}
```

* * * *.

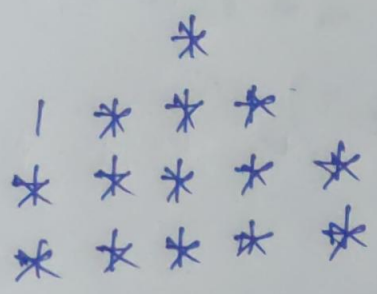
Method 2

```
#include <stdio.h>
int fib(int n)
  {
    int f[n+1];
    int i;
   f[0] = 0;
   f[1] = 1;
for( i=2; i<=n; i++)
   {
    f[i] = f[i-1] + f[i-2];

   }
   return f[n];
  }
  void main( )
```

```
{
  int n;
  scanf("%d",&n);
  printf("%d",fib(n));
  getch()
}.
```

* W.A.P. in C to print patterns of pyramid stars



```
# include <stdio.h>
 void main( )
 { int row, col, n, temp;
printf("Enter the number of rows in pyramid
 of stars\n")
scanf("%d", &n);
 temp = n;
for( row=1; row<=n; row++)
   {
     for( col=1; col<temp; col++)
        { printf(" ");
          temp--;
```

```
    3
for(col=1; col<=2*row-1; col++)
        printf("*");

printf("\n");
 3
   getch();
 3
```

```
  ***
```