Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle,Microsoft
9407077858

LCM (Least Common Multiple)

LCM of two numbers is the smallest number which can be divided by both numbers.

For example, LCM of 15 and 20 is 60, and
LCM of 5 and 7 is 35.

WAP TO FIND LCM

```
#include <stdio.h>
 int    main()
 {
          int n1, n2, max;
          printf("Enter two positive integers\n ");
          scanf("%d%d", &n1, &n2);

         // maximum number between n1 and n2 is stored in max
          max = (n1 > n2) ? n1 : n2 ;

   while (1) //  infinite loop
   {
             if ( max % n1 == 0  &&  max % n2 == 0 )
             {
               printf("The LCM of %d and %d is %d.", n1, n2, max);
               break;
             }
             max++;
   }
}
```

n1 = 2 , n2 = 3

max = (2 > 3) ? 2 : 3 ; F
max = 3

max % n1 == 0 && max % n2 == 0

max = 3 3 % 2 == 0 && 3 % 3 == 0 F

max = 4 4 % 2 == 0 && 4 % 3 == 0 F

max = 5 5 % 2 == 0 && 5 % 3 == 0 F

max = 6 6 % 2 == 0 && 6 % 3 == 0 T break

LCM = max = 6

Highest Common Factor (HCF) : OR
Greatest Common Divisor (GCD) :

The HCF or GCD of two integers is the largest integer

that can exactly divide both numbers (without a remainder).

For example, HCF of 12 and 18 is 6. AND
HCF of 16 and 32 is 16. AND
HCF of 12 and 15 is 3.

WAP to Find HCF OR GCD of two Numbers

```
#include <stdio.h>

    int   main()
    {
           int  n1, n2, i, gcd;

           printf("Enter two integers:\n ");

          scanf("%d%d", &n1, &n2);
          for( i = 1  ;  i <= n1  && i <= n2 ; i++ )
          {
              // Checks if i is factor of both integers

                 if( n1 % i ==0 && n2 % i == 0 )
                          gcd = i;
          }

          printf("G.C.D of %d and %d is %d\n", n1, n2, gcd);

    }
```

HCF of 12 and 18 is 6.

n1 = 12 , n2 = 18

n1 % i ==0 && n2 % i == 0 gcd = i

i = 1   12 % 1 == 0 && 18 % 1 == 0   T   gcd = 1

i = 2   12 % 2 == 0 && 18 % 2 == 0   T   gcd = 2

i = 3   12 % 3 == 0 && 18 % 3 == 0   T   gcd = 3

i = 4   12 % 4 == 0 && 18 % 4 == 0   F

i = 5   12 % 5 == 0 && 18 % 5 == 0   F

i = 6   12 % 6 == 0 && 18 % 6 == 0   T gcd = 6

.
.
.
i = 12   12 % 12== 0 && 18 % 12 == 0   F

LCM * GCD = (N1 * N2)

LCM = (N1 *N2) / GCD ;

WAP TO FIND LCM USING GCD

```
#include <stdio.h>
 int   main()
 {
        int  n1 , n2 , i , min , gcd , LCM ;

       printf("Enter two positive integers:\n ");

       scanf("%d%d", &n1, &n2);

      // minimum number between n1 and n2 is stored in min
        min = (n1 < n2) ? n1 : n2 ;

      for ( i = 1 ; i <= min ;  i++ )
      {
            // check if i is a factor of both integers

            if ( n1 % i == 0 && n2 % i == 0)
                        gcd = i;

      }

      LCM = (n1 * n2) / gcd;

      printf("The LCM   of %d and %d is %d\n" , n1, n2, LCM);

      printf("THE GCD   of %d  and  %d  is  %d\n" , n1, n2 , gcd);

}
```

E.G. OF break STATEMENT

PRIME NO.

```
#include<stdio.h>

    int   main()
    {
         int  i , n , t = 1 ;

         printf(" ENTER NO \n ");
         scanf( "%d", &n);

         for( i = 2 ; i <= n-1  ; i++)
         {
             t = n % i ; // remainder

             if( t == 0 )
             {      break;            }
         }  // imp  for --> end

         if( t != 0 )
         {
                printf(" PRIME NO \n");
         }
         else
         {
                printf(" NOT PRIME NO OR COMPOSITE NO. \n ");
         }
  }
```

-

trace:-

```
1.      n = 5 ;
            for i = 2 to  4

            i = 2   t = 5 % 2 = 1
            i = 3   t = 5 % 3 = 2
            i = 4   t = 5 % 4 = 1 ✓

        if( t != 0 )                PRIME NO.
--------------------------------------------------------------------------------
2.      n = 9
            for i = 2  to  8

            i = 2  t = 9 % 2 = 1
            i = 3  t = 9 % 3 = 0 ✓
              if( t == 0)
                    break ;  i = 4,5,6,7,8 X

if( t != 0 ) X

              NOT PRIME NO. OR COMPOSITE NO.
        ----------------------------------------------------------------------
3.  n = 2 , t = 1    // IMP

        for i = 2  to 1 X

        if( t != 0)   if( t )    // true
              prime no.

--------------------------------------------------------------------------------
```

CONTINUE STATEMENT

WAP FOR PRINT 1 TO 100 NO. EXCEPT 20 TO 30.

```
#include<stdio.h>

        int   main()
        {
                int  i;

                for( i = 1 ; i <= 100 ; i++ )
                {
                      if( i >= 20 && i <= 30 )
                      {
                             CONTINUE;
                      }
                        printf("%d\t" , i);
                }
        }
```

WITHOUT USING CONTINUE STATEMENTS

```
                for( i = 1 ; i <= 100 ; i++ )
                {
                        if( ! (i >= 20 && i <= 30) )
                        {
                                    printf("%d\t" , i);
                        }

                }
```

FIBONACCI SERIES

0 1 1 2 3 5 8 13 .....

given

```
e.g. n = 4        0    1    1     2    3
                 0th  1st  2nd  3rd  4th
```

```
#include<stdio.h>
int    main()
{
            int  i , n , a = 0 , b = 1 , c ;

            printf(" ENTER NO \n ");
            scanf("%d", &n);

            printf("%4d%4d", a , b);  //  0    1

            for( i = 1 ; i <= n-1 ; i++)
            {
                   c = a + b ;
                   printf("%4d", c);

                   a = b ; // imp ( new a and b )
                   b = c ;
            }
 }
```

JUMPING STATEMENT

goto :- TO MOVE PERTICULAR LINE NO.

syntax
goto lebel;

-

BASIC , PASCAL --> GOTO
AVOID GOTO

-

PRINT 1 TO 10 USING if()

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

         p :   printf(" %d\n",i);

                i++;

                if( i <= 10 )
                {
                      goto  p ;
                }
    }
```