**ASSIGNMENT NUMBER: - 01**

**PROBLEM STATEMENT:**

**Program in C to print all the prime factors of a number in descending order.**

**THEORY: -** A **Prime Number** is any number that is only divisible by itself and 1. Some examples of prime numbers include 2, 5 and 17.

To **factor** a number is to break that number down into smaller parts. For example, factors of 8 could be 2 and 4 because 2 \* 4 is 8. The factors of 24 are 1, 2, 3, 4, 6, 8, 12 and 24, because 1 \* 24 is 24, 2 \* 12 is 24, 3 \* 8 is 24 and so is 4 \* 6. So all of these numbers are said to be factors of 24.

To find the **prime factorization** of a number, we need to break that number down to its prime factors. To determine the prime factors, first we need to calculate all the factors of that number and then check which of those factors are not divided with a number between 2 to the square root of that factor. The factors which shows such property are the prime factors of the number.

**EXAMPLE**: - The factors of 20 are 1,2,4,5,10. But 4 and 10 are again divisible by a number other than 1 and itself. So only 2 and 5 are prime factors.

**ALGORITHM: -**

**Input: The number whose prime factor is to be calculated, say variable n.**

**Output: The prime factors of the number in descending order, if any.**

**Steps:**

Step 1: Print ”Prime factors of n are: ”

Step 2: Repeat through Step 3 to Step 10 for i= n to 2

Step 3: if(n mod i=0)

Then

Step 4: Set p=1

Step 5: Repeat through Step 6 to Step 8 for j= i /2 to 2

Step 6: If(I mod j=0)

Then

Step 7: Set p=0

Break

**[** End of inner if structure **]**

Step 8: Set j=j-1

**[** End of inner for loop **]**

Step 9: if(p=1)

Then

Print “ i “

**[** End of if structure **]**

**[** End of outer if structure **]**

Step 10: Set i=i-1

**[** End of for loop **]**

Step 11**: End.**

**SOURCE CODE: -**

#include<stdio.h>

int main()

{

int n,i,j,p,ch;

printf("\nEnter the number :");

scanf("%d",&n);

printf("\nPrime factors of %d are: ",n);

for(i=n;i>=2;i--)

{

if(n%i==0) // ‘*i’ is the factor of ‘n’*

{

p=1;

for(j=i/2;j>=2;j--)

{

if(i%j==0) // *checking if factors of n are prime or not*

{

p=0; // ‘*i’ is not a prime factor of n*

break; // *exit from condition*

}

}

if(p==1)

{

printf(" %d ",i); // *printing the prime factors*

}

}

}

return 0;

}

**INPUT & OUTPUT: -**

**SET 1:** Enter the number: 10

The prime factors in descending order: 5,2

**SET 2**: Enter the number: 38

The prime factors in descending order: 19,2

**DISCUSSION: -** The above program is a basic method which gives the prime factors of a given number. The computational time is less for smaller numbers where number of iterations is less. But for a larger number, as there is more number of iterations, more is the time complexity. So this Method is not convenient for larger numbers.