Tribhuvan University
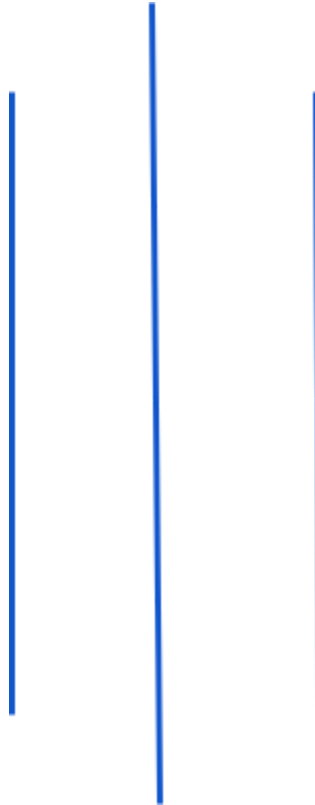## Institute of Engineering
Thapathali Campus,Thapathali

LAB SHEET #3

Submitted by:
Name: Kishan Adhikari
Roll No. :THA077BCT021

Submitted to:
Department of Electronics and Computer Engineering

Date : 31st July 2021

Title:

Write a program to find sum as Y of the following series excluding prime numbers in the series.

$$Y = 1 + \frac{1}{1!} + \frac{2^2}{2!} + \frac{3^2}{3!} + \cdots + \frac{10^2}{10!}$$

Problem Analysis:

The problem is to find the sum of given series excluding prime numbers. To find the sum ,we first declare the fact(int) function to calculate factorial to a number. The variable declared are sum(float), prime_sum(float),i(int),j(int),count(int).

$5! = 5 * 4 * 3 * 2 * 1$

To calculate factorial of a number use of loop from n to 1 and calculating factorial as:

$fact = fact * n$

We then use loop to calculate sum from 1 to 10 as : sum+= (i*i)/fact(i).

Now subtraction of prime numbers in a given series is yet to remain. To calculate sum of prime in series: check whether a number is prime or not and if it is prime then prime_sum+=(j*j)/fact(j) where j is prime number. Use of count variable to keep track of prime numbers . If the number is prime count=0 else count=1.
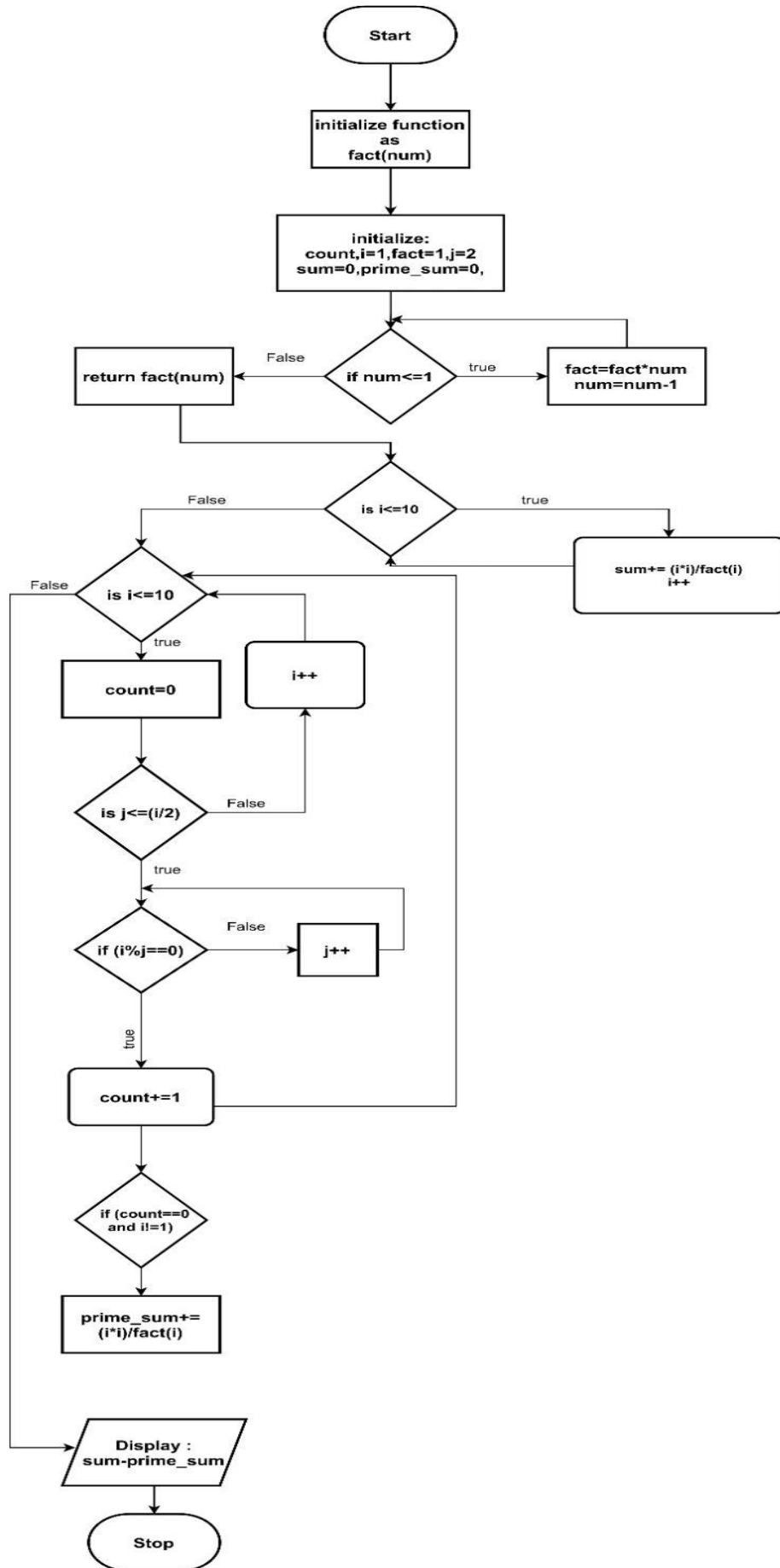
Required_sum is calculated as:

$Required\_sum = sum - prime\_sum$

Input variables	Processing variables/calculations	Output variables	Necessary header files/function
sum(float) fact(int) i(int) j(int) count(int) prime_sum(float)	sum+= $(i^2)/facto(i)$ prime_sum+= $(i^2)/facto(i)$ (i%j)		stdio.h facto()

Algorithm:

1. Start
2. Declare fact(num) function
3. facto=1,initialize num,i,count,sum=0,prime_sum=0
4. If num<0 : Factorial does not exists
5. If num==0; facto=1
6. Repeat till num>=1 true facto*=num
7. If statement 5 is false return facto
8. i =1 to 10 goto 9
9. sum+=(i*i)/fact(i) goto 8 until i<=10
10. j=1 to 10
11. count=0
12. k=2 to 5 goto 13
13. (j%k==0) true ;count++ ;break false goto 14
14. (count==0 and i!=1) true goto 15
15. prime_sum+= (i*i)/fact(i)
16. required sum=sum-prime_sum
17. Display required sum
18. stop

Flowchart:

```
                          ┌─────────┐
                          │  Start  │
                          └────┬────┘
                               │
                               ▼
                     ┌───────────────────┐
                     │ initialize function│
                     │         as         │
                     │      fact(num)     │
                     └─────────┬─────────┘
                               │
                               ▼
                     ┌───────────────────┐
                     │     initialize:    │
                     │  count,i=1,fact=1,j=2│
                     │  sum=0,prime_sum=0, │
                     └─────────┬─────────┘
                               │
                               ▼
         False          ◇ if num<=1 ◇          true      ┌──────────────┐
  ┌──────────────┐ ◄──────                  ──────────►  │ fact=fact*num│
  │return fact(num)│                                     │  num=num-1   │
  └──────┬───────┘                                       └──────────────┘
         │
         ▼
   False ◇ is i<=10 ◇  true
  ◄──────                  ──────────►  ┌──────────────┐
                                        │ sum+= (i*i)/fact(i)│
                                        │       i++    │
                                        └──────────────┘
  False
  ◄─── ◇ is i<=10 ◇
              │ true
              ▼
        ┌──────────┐       ┌──────┐
        │ count=0  │       │ i++  │
        └────┬─────┘       └──────┘
             │
             ▼
        ◇ is j<=(i/2) ◇  False
              │ true
              ▼
        ◇ if (i%j==0) ◇  False   ┌──────┐
              │ true             │ j++  │
              ▼                  └──────┘
        ┌──────────┐
        │ count+=1 │
        └────┬─────┘
             │
             ▼
        ◇ if (count==0 ◇
        │  and i!=1)  │
             │
             ▼
        ┌──────────────┐
        │  prime_sum+= │
        │  (i*i)/fact(i)│
        └──────────────┘
             │
             ▼
        ╱ Display :     ╱
       ╱ sum-prime_sum ╱
             │
             ▼
        ┌─────────┐
        │  Stop   │
        └─────────┘
```

Source Code:

```c
/*
 Author: Kishan Adhikari
 Created Date: 2078/04/30
 Filename:sumwithoutprime.c
 Description:program to print sum of series excluding prime numbers
 series : Y=1+1/(1!)+(2^2/2!)+......+(10^2/10!)
*/

#include <stdio.h>

int fact(int num)
{
 int facto = 1;
 if (num < 0)
 {
   printf("Factorial of negative doesnt exists\n");
 }
 if (num == 0)
 {
   return 1;
 }

 for (num; num >= 1; num--)
 {
   facto = facto * num;
 }
 return facto;
}

int main()
{

 int num, i, count;
 float sum = 0, prime_sum = 0;
 for (int j = 1; j <= 10; j++)
 {
   sum += (float)(j * j) / fact(j);
 }
 sum += 1;
```

```c
    for (num = 1; num <= 10; num++)
    {

        count = 0;

        for (i = 2; i <= num / 2; i++)
        {
            if (num % i == 0)
            {
                count++;
                break;
            }
        }

        if (count == 0 && num != 1)
            prime_sum += (float)(num * num) / fact(num);
    }

    printf("Sum of series excluding prime number is: %.2f\n", sum -
prime_sum);

    return 0;
}
```
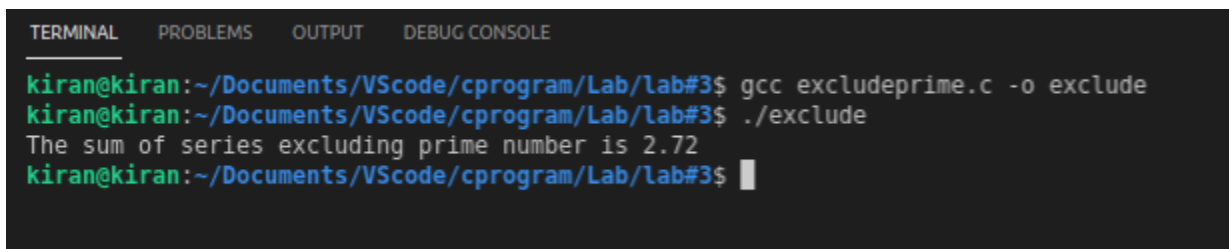
Output:

```
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ gcc sumwithoutprime.c -o sumwithout
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ./sumwithout
Sum of series excluding prime number is: 2.72
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ []
```

Title:

Write a program to find the sum as Y of the following series excluding prime numbers in the series. (Write a function program to check whether the number is prime or not. also write a recursive function to calculate the factorial of the series numbers).

$$Y = 1 + \frac{1}{1!} + \frac{2^2}{2!} + \frac{3^2}{3!} + \cdots + \frac{10^2}{10!}$$

Source Code:

```c
/*
 @Author : Kishan Adhikari
 @File name: excludeprime.c
 @Created Date:2021/04/30
 @Description: program to find sum as Y of the following series excluding
prime number in the series. (Write function program to check whether the
number is prime or not. also write recursive function to calculate the
factorial of the series numbers)
 series : Y=1+1/(1!)+(2^2/2!)+......+(10^2/10!)
*/

#include <stdio.h>
int facto(int num)
{
 if (num <= 0)
 {
    return 1;
 }
 return num * facto(num - 1);
}

int check_prime(int num)
{
 int count;
 for (int i = 2; i <= (num / 2); i++)
 {
```

```c
      count = 0;
      if (num % i == 0)
      {
        count++;
        break;
      }
    }
    if (count == 0)

      printf("%d is a prime number.\n", num);
    else
    {
      printf("%d is a composite number.\n", num);
    }
}
float sum_of_prime()
{
  float prime_sum = 0;
  int i, j, count;
  for (i = 1; i <= 10; i++)
  {
    count = 0;

    for (j = 2; j <= (i / 2); j++)
    {
      if (i % j == 0)
      {
        count++;
        break;
      }
    }
    if (count == 0 && i != 1)
    {
      prime_sum += (float)(i * i) / facto(i);
    }
  }
  return prime_sum;
}

int main()
```

```c
{
  float sum = 0;
  for (int i = 1; i <= 10; i++)
  {
    sum += (float)(i * i) / facto(i);
  }
  sum += 1;

  printf("Sum of series excluding prime number is %.2f\n", sum -
sum_of_prime());
  return 0;
}
```

Output:

Discussion and Conclusion:

In this way the sum of given series by excluding the prime number was calculated. From this lab I understood the use of For loops, Nested for loops,Recursion, User-defined function and problem solving skill using c language.

Title:

Write a program to input two integer numbers and display the sum of even numbers between these two input numbers.

Source Code:
```c
/*
 @Author : Kishan Adhikari
 @File name: sumofeven.c
 @Created Date:2021/04/30
 @Description: program to print sum of even number
```

```c
 between two input numbers
*/

#include <stdio.h>
int main()
{
 int num1, num2, sum = 0;
 printf("Enter two numbers:\n");
 scanf("%d%d", &num1, &num2);
 int temp1 = num1 + 1; //incrementing num1 by 1 as num1 doesn't lie
between(num1,num2)
 if (num1 < num2)
 {
    for (temp1; temp1 < num2; temp1++)
    {
      if (temp1 % 2 == 0)
      {
        sum += temp1;
      }
    }
    printf("Sum=%d\n", sum);
 }

 else
 {
    int temp2 = num2 + 1; //incrementing num2 by 1 as num2 doesn't lie
between(num2,num1)
    for (temp2; temp2 < num1; temp2++)
    {
      if (temp2 % 2 == 0)
      {
        sum += temp2;
      }
    }
    printf("Sum=%d\n", sum);
 }
 return 0;
}
```

Output:

Title:

Write a program to find GCD (greatest common divisor or HCF) and LCM (least common multiple) of two numbers.

Source Code:

```
/*
 @Author : Kishan Adhikari
 @File name: GCDandLCM.c
 @Created Date:2021/04/31
 @Description: program to find GCD (greatest common divisor or HCF) and
LCM (least common
 multiple) of two numbers.

 GCD (Greatest common divisor) of two number is largest number
 which divides boths
 gcd(120,250) :
 to calculate divisor we find all the divisor of small number (120)
 divide number (120) by  2 to 10
 {1,120} {2,60} {3,40} {4,30} {5,24} {6,20} {8,15} {10,12}
 and we move from bigger number to check if it evenly divides 250 or not
 in our case 120 doesnt,60 doesnt,40 doesnt,30 doesnt,24 doesnt,20
doesnt,15 doesnt,12doesnt ,10 does
 it takes 19 steps to calculate
 gcd(120,250)=10
 To calculate GCD we use Euclidian Algorithm:
```

```c
 GCD(120,250)
 divide 250 by 120 and save remainder
 250%120=10
 divide 120 by remainder(10)
 120%10=0
 since remainder is zero ,gcd is 10


 */
#include <stdio.h>

int GCD(int num1, int num2);
int lcm(int a, int b);

int main()
{
 int a, b;
 printf("Enter Value of a and b: \n");
 scanf("%d%d", &a, &b);

 printf("The GCD of %d and %d is: %d\n", a, b, GCD(a, b));
 printf("The LCM of %d and %d is : %d\n", a, b, lcm(a, b));
 return 0;
}
int GCD(int num1, int num2)
{
 int dividend = num1 >= num2 ? num1 : num2;
 int divisor = num1 <= num2 ? num1 : num2;
 while (divisor != 0)
 {
   int remainder = dividend % divisor;
   dividend = divisor;
   divisor = remainder;
 }
 return dividend;
}
int lcm(int a, int b)
{
 int LCM = a * b / (GCD(a, b));
 return LCM;
}
```

Output:

Title:

Write a program to display Fibonacci series of last term up to 300.

Source Code:

```c
/*
 @Author : Kishan Adhikari
 @File name: lastfibo.c
 @Created Date:2021/04/30
 @Description: program to display Fibonacci series of last term up to 300.
*/
#include <stdio.h>
int main()
{

 int end = 300, i, c, last;
 int a = 0;
 int b = 1;

 for (i = 0; i < 25; i++)
 {
   c = a + b;
   a = b;
   b = c;
   if (c >= end)
   {
```

```c
        last = a;


        break;
    }
}
printf("The last fibonacci term less than 300 is :%d\n", last);
}
```

Output:


```
TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE

kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ gcc lastfibo.c -o last
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ./last
The last fibonacci term less than 300 is :233
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ▊
```

Title:
Write a program to display the flag of Nepal using symbolic/HEX character in C.
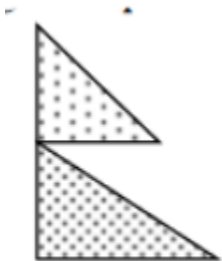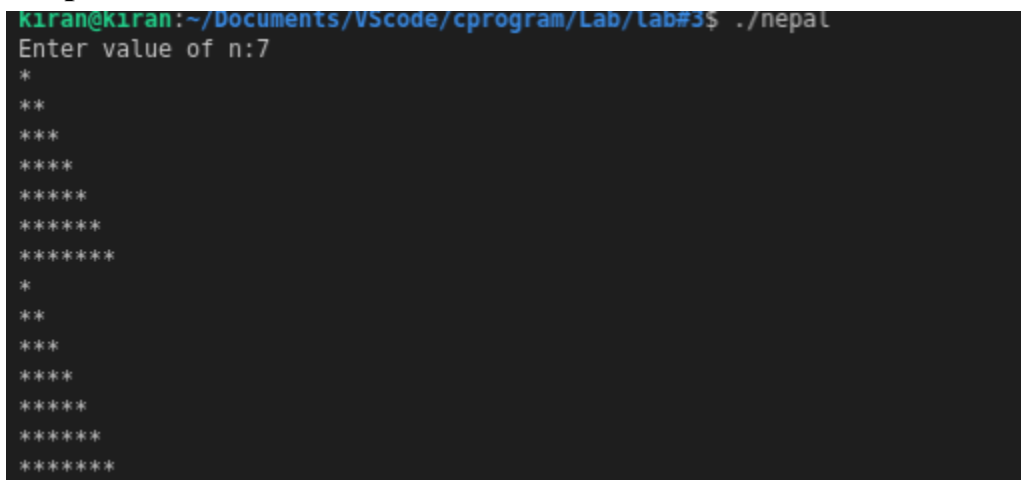
Source Code:
```c
/*
 @Author : Kishan Adhikari
 @File name: nepal.c
 @Created Date:2021/04/30
 @Description: program to flag of nepal.
*/
#include <stdio.h>
```

```c
int main()
{
 int i, j, n;
 printf("Enter value of n:");
 scanf("%d", &n);
 for (int k = 0; k < 2; k++)
 {
   for (i = 0; i < n; i++)
   {
     for (j = 0; j <= i; j++)
     {
       printf("*");
     }
     printf("\n");
   }
 }
 return 0;
}
```

Output:

```
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ./nepal
Enter value of n:7
*
**
***
****
*****
******
*******
*
**
***
****
*****
******
*******
```

Title:

Write a program to add, subtract, multiply and divide two integers using user defined type function with return type.

Source Code:

```c
/*
 @Author : Kishan Adhikari
 @File name: userdefined.c
 @Created Date:2021/04/30
 @Description: program to add, subtract, multiply and divide two integers
using user defined type
 function with return type..
*/
#include <stdio.h>
int add(int num1, int num2);
int subtract(int num1, int num2);
int multiply(int num1, int num2);
float divide(int num1, int num2);
int main()
{
 int a, b;
 printf("Enter two numbers:\n");
 scanf("%d%d", &a, &b);
 printf("The value of a+b is: %d\n", add(a, b));
 printf("The differences of a and b is: %d\n", subtract(a, b));
 printf("The Product of a and b is: %d\n", multiply(a, b));
 printf("The Value of a/b is: %.2f\n", divide(a, b));

 return 0;
}
int add(int num1, int num2)
{
 return num1 + num2;
}
int subtract(int num1, int num2)
{
 return num1 - num2;
}
int multiply(int num1, int num2)
{
```

```c
    return num1 * num2;
}
float divide(int num1, int num2)
{
    return (float)num1 / num2;
}
```

Output:

```
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ gcc userdefined.c -o user
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ./user
Enter two numbers:
4
8
The value of a+b is: 12
The differences of a and b is: -4
The Product of a and b is: 32
The Value of a/b is: 0.50
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$
```

Title:

Write a program to calculate sum of first 50 natural numbers using recursive function.

Source Code:

```c
/*
 @Author : Kishan Adhikari
 @File name: userdefined.c
 @Created Date:2021/04/30
 @Description: program to calculate sum of first 50 natural numbers using
recursive function.

 logic:
 sum(50)=50+sum(49)

 formula :n*(n+1)/2 = 50*51/3=25*51=1275

*/

#include <stdio.h>
int recrsum(int num)
{
```

```c
    if (num == 1)
    {
        return 1;
    }
    return num + recrsum(num - 1);
}
int main()
{

    printf("The sum of natural number upto 50 is:%d\n", recrsum(50));
    return 0;
}
```

Output:

Title:
Define a function named fact() to calculate factorial of a number n and then write a program that uses this function fact() to calculate combination and permutation.

Source Code:

```c
/*
 Author: Kishan Adhikari
 Created Date: 2078/04/30
 Filename:permcomb.c
 Description:Define a function named fact() to calculate factorial of a
number n and then write a program
 that uses this function fact() to calculate combination and permutation.
*/

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

```c
    if (num < 0)
    {
        printf("Factorial of negative doesnt exists\n");
    }
    if (num == 0)
    {
        return 1;
    }

    for (num; num >= 1; num--)
    {
        facto = facto * num;
    }
    return facto;
}
int permut(int num, int r)
{

    return fact(num) / fact(num - r);
}
int combt(int num, int r)
{

    return fact(num) / (fact(num - r) * fact(r));
}

int main()
{
    int num1, num2;
    printf("Enter value of n and r:\n");
    scanf("%d%d", &num1, &num2);
    printf("The value of %d permutation %d is %d\n", num1, num2, permut(num1,
num2));
    printf("The value of %d combination %d is %d\n", num1, num2, combt(num1,
num2));
    return 0;
}
```

Output:

Title:

Write a recursive function to generate Fibonacci series.

Source Code:

```c
/*
 Author: Kishan Adhikari
 Created Date: 2078/04/28
 Filename:fibo.c
 Description:recursive function to print fibonacci series
*/

#include <stdio.h>

int fibo(int n)
{
 if (n <= 1)
 {
    return n;
 }
 return fibo(n - 1) + fibo(n - 2);
}
int main()
{
 int num;
 printf("Enter upto which you want fibonacci series\n");
 scanf("%d", &num);
 for (int i = 0; i < num; i++)
 {
    printf("%d\n", fibo(i));
 }
```
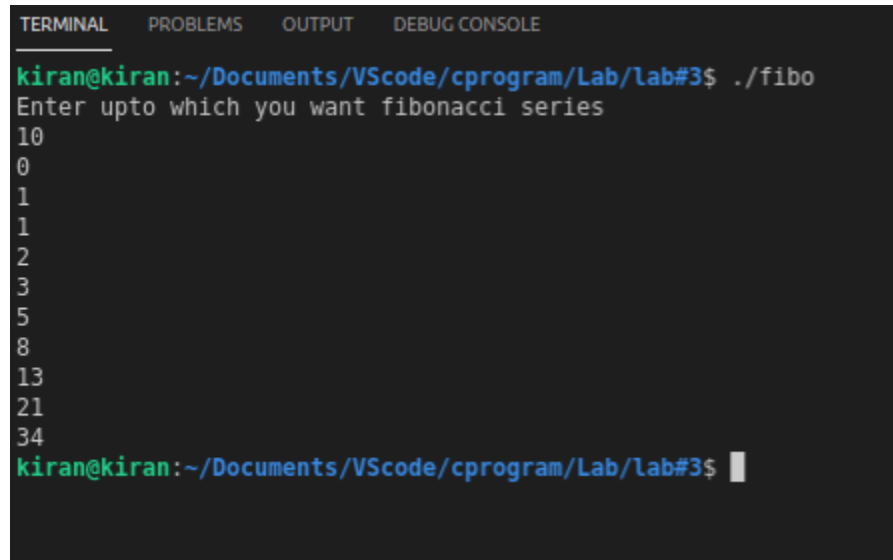
```
    return 0;
}
```

Output:

Title:
Write a program that illustrates use of local, global and static variables

Source Code:

```c
/*
 Author : Kishan Adhikari
 Filename :var.c
 Created Date: 2078/04/29
 Description: Program to illustrate differences between static,global and
local variables.
  local variables: The variables which are declared inside the function,
compound statement (or block) are called Local variables.
 global variables: The variables declared outside any function are called
global variables.
 static variables: A Static variable is able to retain its value between
different function calls.
*/
```

```c
#include <stdio.h>
int c = 250; //global variables as it is defined outside any function

void add();
void subtract();
int main()
{
 add();
 add();
 subtract();
 subtract();
}
void add()
{
 c = 200; //c is global variable and can be accessed by any function
 printf("The value of c=%d\n", c);
 int a = 100; //variable declared inside a function is called local
variables. It cannot be accessed by other functions
 printf("The value of a=%d\n", a);
 c += 50;
}
void subtract()
{
 c = 35; //c is global variable and can be accessed by any function
 printf("The value of c=%d\n", c);
 static int d = 15; //d is static variable and value is attained in each
function call
 printf("The value of d=%d\n", d);
 d++;
}
```

Output:

Title:

Write a program to display the following.

b.

```
1
1    4
1    4    9
1    4    9    16
1    4    9    16    25
```

Source Code:

```c
/*
 Author: Kishan Adhikari
 Filename: squarepattern.c
 created date: 2078/04/25
 Description: Program to print following pattern
 1
 1      4
 1      4      9
 1      4      9      16
 1      4      9      16      25
*/

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

```c
    {
        printf("%d\t", j * j);
    }
    printf("\n");
  }
 }
```

Output:

```
TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE

kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ ./sq
1
1       4
1       4       9
1       4       9       16
1       4       9       16      25
kiran@kiran:~/Documents/VScode/cprogram/Lab/lab#3$ []
```

Title:

Write a program to display the following.

   c.

```
*

*       *

*       *       *

*       *       *       *
```

Source Code:

```
/*
Author: Kishan Adhikari
Filename:ladder.c
Created Date: 2078/04/28
Description:
program to print pattern:
*
```

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

Nested loop is used for such task (pattern with row and column)
Inner loop print how many * to print on single line
Outer loop gives how many such lines are needed.

To print given pattern


     i=1 * j=1(one star)
     i=2 ** j=2(2 star)
     i=3 *** j=3(3 star)
     i=4 **** j=4(4 star)
     i=5 ***** j=5(5 star)
first inner loop completes and moves to outer loop.
there is next life after i=j so we moves to next line after completing
inner loop


                    i<5,j<=i
i          j        condition         result                 printstatement
0          0        0<5,0<=0           true                        *
0          1        0<5,1<=0           false(loop breaks)         \n
1          0        1<5,0<=1           true                        *
1          1        1<5,1<=1           true                        **
1          2        1<5,2<=1           false(loop breaks)         \n


*/



#include <stdio.h>

int main()
{
 int i, j;
 for (i = 0; i < 5; i++)
  {
```

```c
    for (j = 0; j <= i; j++)
    {

      printf("*");
    }
    printf("\n");
 }
}
```

Output:

Title:

Write a program to print following series.

u.

1	6	10	13	15
2	7	11	14	
3	8	12		
4	9			
5				

Source Code:

```c
/*
 Author: Kishan Adhikari
 Filename: pattern.c
 Created Date: 2078/04/27
 Description: program to print following series
 1   6   10 13 15
```

```
 2   7   11 14

 3   8   12

 4   9

 5
*/

#include <stdio.h>

int main()
{
 int i, j, k;
 for (i = 1; i <= 5; i++)
 {
   k = i;

   for (j = 5; j >= i; j--)
   {
     printf("%-3d", k);
     k = k + j;
   }
   printf("\n");
 }
}
```

Output:


```
kiran@kiran:~/Documents/VScode/cprogram/Lab/Lab#3$ ./pattern
1  6   10 13 15
2  7   11 14
3  8   12
4  9
5
```