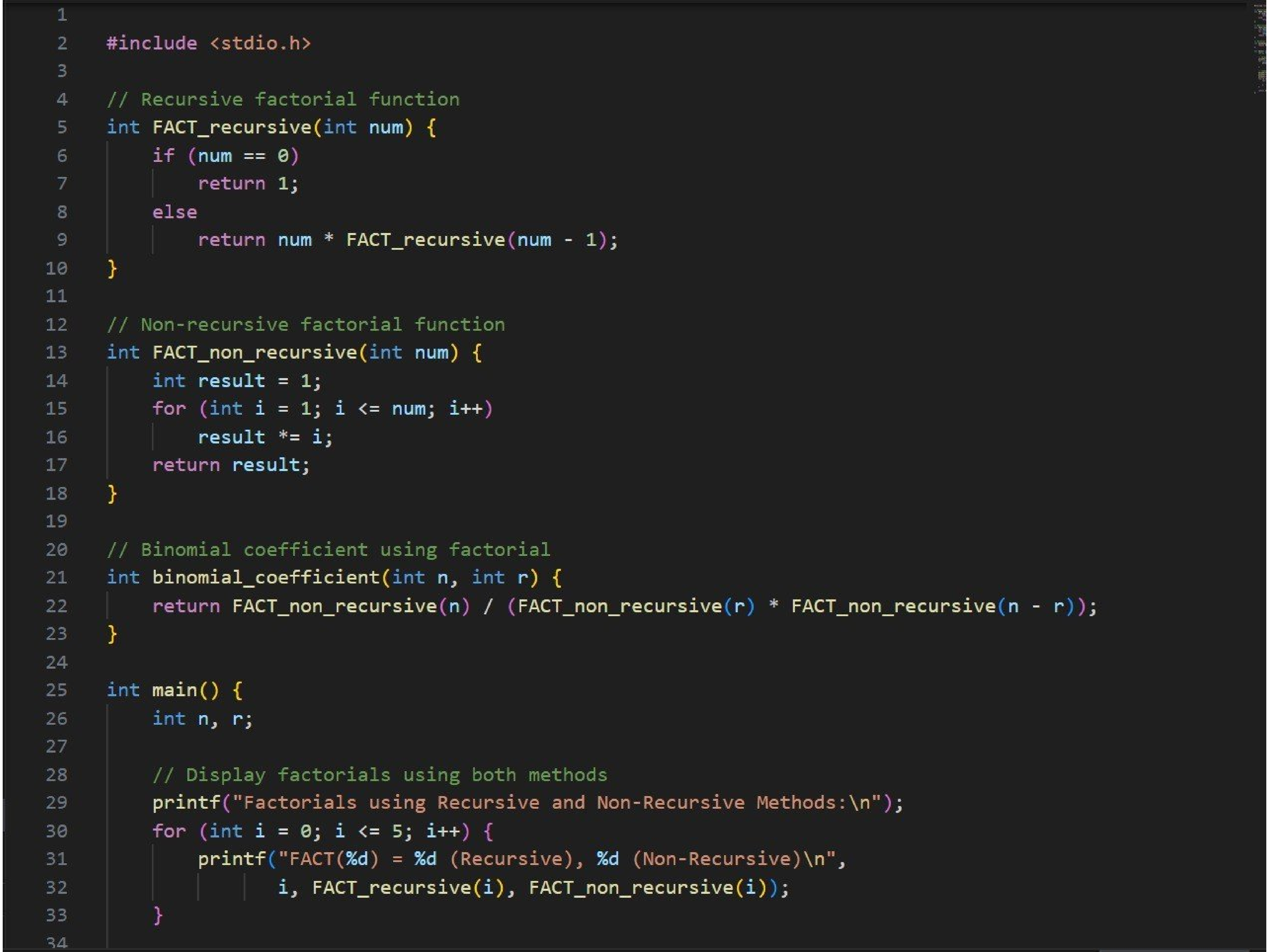
Experiment 6 : Functions.

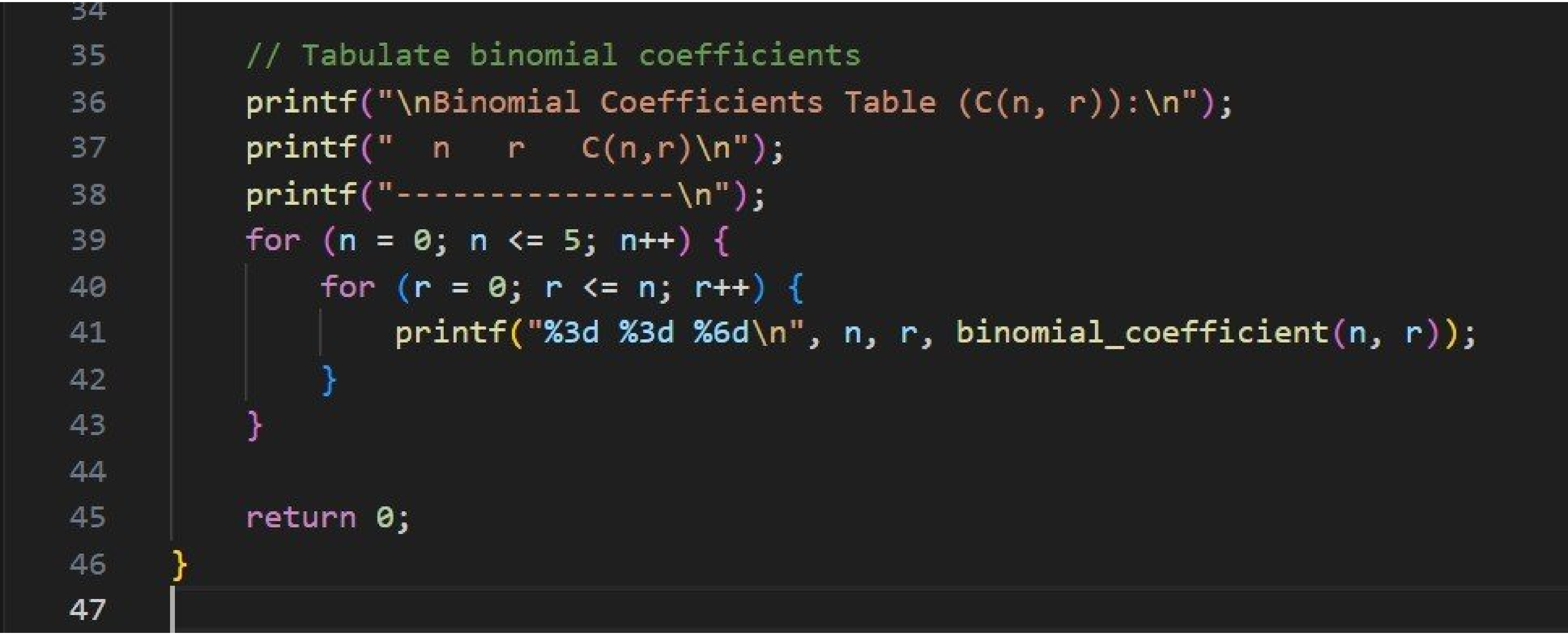
1. Develop a recursive and non-recursive function FACT (num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n = 0. Otherwise, FACT(n) = n * FACT (n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages.

```
#include <stdio.h>
 int FACT_recursive (int num) {
     if (num == 0)
      return 1;
     else
       return   num * FACT_recursive (num-1);
      }
     int FACT_non_recursive (int num) {
      int result = 1;
      for ( int i = 1; i <= num ; i++)
       result *= i ;
      return result ;
     }
     int binomial_coefficient (int n, int r) {
        return FACT_non_recursive (n) / (FACT_non-recursive (r) * FACT_non-recursive (n-r))
      }
    int main () {
      int n, r ;
      printf ("Factorials using Recursive and non-recursive methods : \n");
      for (int i = 0; i <= 5 ; i++) {
```

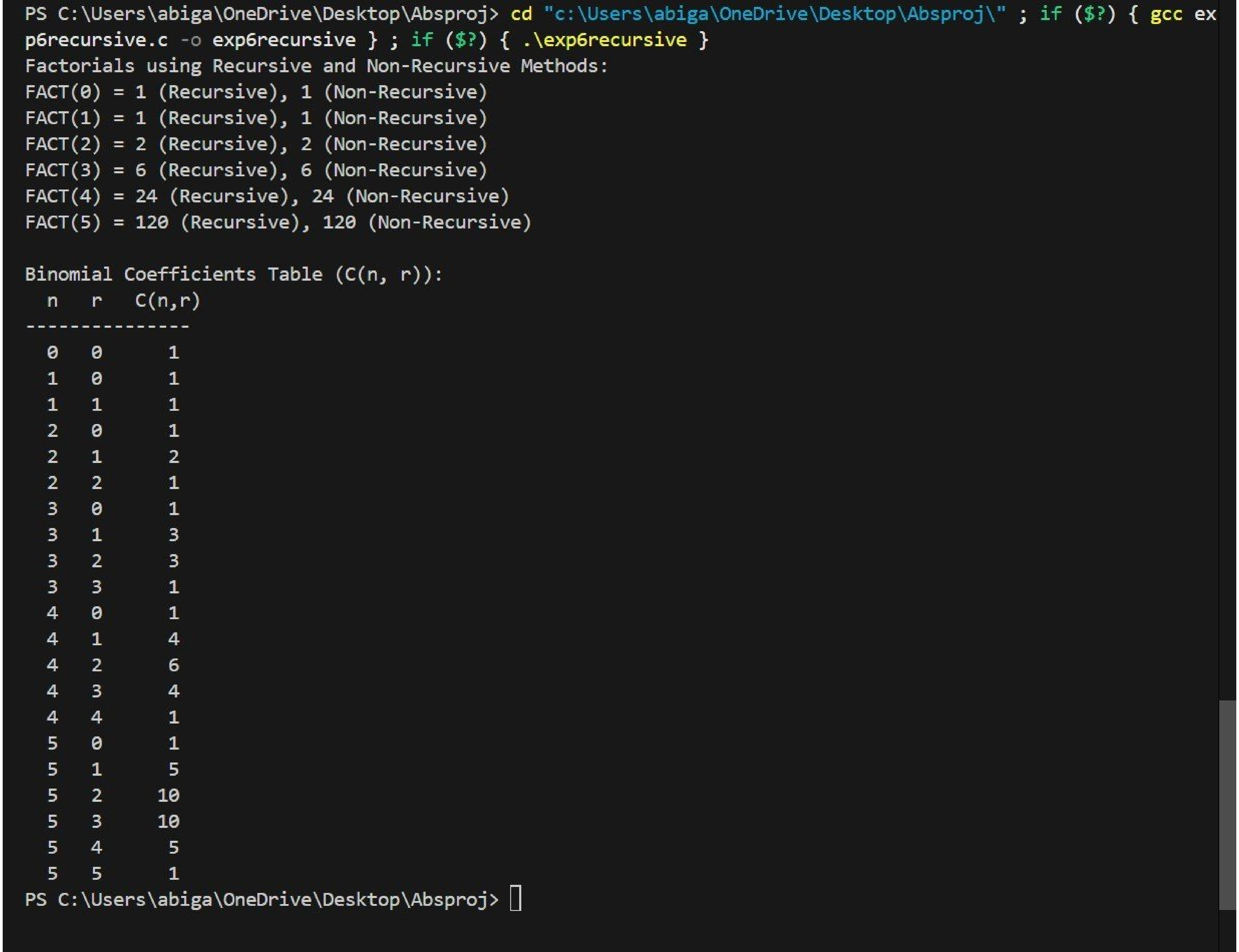
```
printf ("FACT (%d) = %d (Recursive), %d (Non-recursive) \n", i,
        FACT_recursive (i), FACT_non-recursive (i));
 }
  printf ("\n Binomial Coefficients Table (C(n,r)):\n");
  printf ("  n   r    C(n,r) \n");
  printf ("-----------------\n");
  for (n=0 ; n<=5 ; n++) {
      for (r= 0 ; r<=n ; r++) {
          printf ("%3d %3d %6d \n", n, r, binomial-coefficient (n,r));
      }
   }

  return 0;
 }
```



```c

#include <stdio.h>

// Recursive factorial function
int FACT_recursive(int num) {
    if (num == 0)
        return 1;
    else
        return num * FACT_recursive(num - 1);
}

// Non-recursive factorial function
int FACT_non_recursive(int num) {
    int result = 1;
    for (int i = 1; i <= num; i++)
        result *= i;
    return result;
}

// Binomial coefficient using factorial
int binomial_coefficient(int n, int r) {
    return FACT_non_recursive(n) / (FACT_non_recursive(r) * FACT_non_recursive(n - r));
}

int main() {
    int n, r;

    // Display factorials using both methods
    printf("Factorials using Recursive and Non-Recursive Methods:\n");
    for (int i = 0; i <= 5; i++) {
        printf("FACT(%d) = %d (Recursive), %d (Non-Recursive)\n",
               i, FACT_recursive(i), FACT_non_recursive(i));
    }
```



```c

    // Tabulate binomial coefficients
    printf("\nBinomial Coefficients Table (C(n, r)):\n");
    printf("  n   r   C(n,r)\n");
    printf("---------------\n");
    for (n = 0; n <= 5; n++) {
        for (r = 0; r <= n; r++) {
            printf("%3d %3d %6d\n", n, r, binomial_coefficient(n, r));
        }
    }

    return 0;
}

```



```
PS C:\Users\abiga\OneDrive\Desktop\Absproj> cd "c:\Users\abiga\OneDrive\Desktop\Absproj\" ; if ($?) { gcc ex
p6recursive.c -o exp6recursive } ; if ($?) { .\exp6recursive }
Factorials using Recursive and Non-Recursive Methods:
FACT(0) = 1 (Recursive), 1 (Non-Recursive)
FACT(1) = 1 (Recursive), 1 (Non-Recursive)
FACT(2) = 2 (Recursive), 2 (Non-Recursive)
FACT(3) = 6 (Recursive), 6 (Non-Recursive)
FACT(4) = 24 (Recursive), 24 (Non-Recursive)
FACT(5) = 120 (Recursive), 120 (Non-Recursive)

Binomial Coefficients Table (C(n, r)):
  n   r   C(n,r)
---------------
  0   0      1
  1   0      1
  1   1      1
  2   0      1
  2   1      2
  2   2      1
  3   0      1
  3   1      3
  3   2      3
  3   3      1
  4   0      1
  4   1      4
  4   2      6
  4   3      4
  4   4      1
  5   0      1
  5   1      5
  5   2     10
  5   3     10
  5   4      5
  5   5      1
PS C:\Users\abiga\OneDrive\Desktop\Absproj> 
```