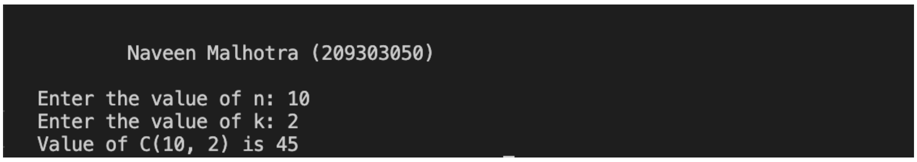
LAB 8 : Dynamic Programming

1. Binomial Coefficient

```
#include <stdio.h>
#include <stdlib.h>

int min(int a, int b);

int binomialCoeff(int n, int k)
{
  int C[n + 1][k + 1];
  int i, j;

  for (i = 0; i <= n; i++)
  {
    for (j = 0; j <= min(i, k); j++)
    {
      if (j == 0 || j == i)
        C[i][j] = 1;

      else
        C[i][j] = C[i - 1][j - 1] + C[i - 1][j];
    }
  }

  return C[n][k];
}
int min(int a, int b)
{
  return (a < b) ? a : b;
}
int main()
{
  printf("\n\n\tNaveen Malhotra (209303050)\n\n");
  int n, k;
  printf("Enter the value of n: ");
  scanf("%d", &n);
  printf("Enter the value of k: ");
  scanf("%d", &k);
  printf("Value of C(%d, %d) is %d ", n, k, binomialCoeff(n, k));
  printf("\n");

  return 0;
}
```

OUTPUT:

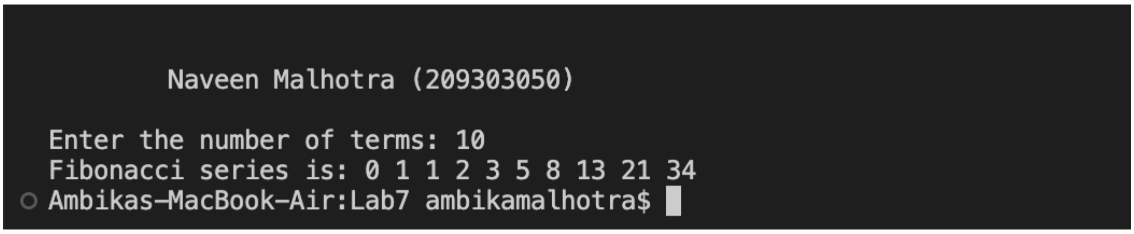


```
        Naveen Malhotra (209303050)

Enter the value of n: 10
Enter the value of k: 2
Value of C(10, 2) is 45
```

2. Fibonacci Series

```
#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];
}
int main()
{
   printf("\n\n\tNaveen Malhotra (209303050)\n\n");
   int n;
   printf("Enter the number of terms: ");
   scanf("%d", &n);
   printf("Fibonacci series is: ");
   for (int i = 0; i < n; i++)
   {
      printf("%d ", fib(i));
   }
   printf("\n");
   return 0;
}
```

OUTPUT :



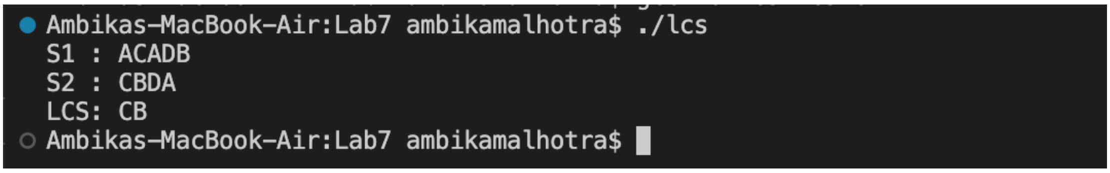
```


        Naveen Malhotra (209303050)

  Enter the number of terms: 10
  Fibonacci series is: 0 1 1 2 3 5 8 13 21 34
○ Ambikas-MacBook-Air:Lab7 ambikamalhotra$ █
```

3 . LCS : Longest Common Sequence

```
#include <stdio.h>
#include <string.h>

int i, j, m, n, LCS_table[20][20];
char S1[20] = "ACADB", S2[20] = "CBDA", b[20][20];

void lcsAlgo()
{
   m = strlen(S1);
   n = strlen(S2);

   for (i = 0; i <= m; i++)
      LCS_table[i][0] = 0; // Filling the first row and column as 0
   for (i = 0; i <= n; i++)
      LCS_table[0][i] = 0;

   for (i = 1; i <= m; i++) // Starting from the row with index 1
      for (j = 1; j <= n; j++)
      {
         if (S1[i - 1] == S2[j - 1]) // If the caracters are same
         {
            LCS_table[i][j] = LCS_table[i - 1][j - 1] + 1; // Diagonal value + 1
         }
         else if (LCS_table[i - 1][j] >= LCS_table[i][j - 1]) // if value above is greater than select that
         {
            LCS_table[i][j] = LCS_table[i - 1][j];
         }
         else // if tha value is
         {
            LCS_table[i][j] = LCS_table[i][j - 1];
         }
      }

   int index = LCS_table[m][n];
   char lcsAlgo[index + 1];
   lcsAlgo[index] = '\0';

   int i = m, j = n;
   while (i > 0 && j > 0)
   {
      if (S1[i - 1] == S2[j - 1])
      {
         lcsAlgo[index - 1] = S1[i - 1];
         i--;
         j--;
         index--;
      }

      else if (LCS_table[i - 1][j] > LCS_table[i][j - 1])
         i--;
```

```
        else
            j--;
    }

    // Printing the sub sequences
    printf("S1 : %s \nS2 : %s \n", S1, S2);
    printf("LCS: %s", lcsAlgo);
}

int main()
{
    lcsAlgo();
    printf("\n");
}
```

OUTPUT :



```
● Ambikas-MacBook-Air:Lab7 ambikamalhotra$ ./lcs
  S1 : ACADB
  S2 : CBDA
  LCS: CB
○ Ambikas-MacBook-Air:Lab7 ambikamalhotra$ █
```