LCS

```
#include<bits/stdc++.h>

 using namespace std;

  int main()

 {
     string s1,s2;

     cin >> s1 >> s2;

     int n = s1.size(),m = s2.size();

     int lcs [n+1] [m+1];

     char dir [n+1] [m+1];

     for ( int i =0; i<=n; i++)

       lcs [i][0] = 0;

       for ( int j=0; j<=m; j++)

       lcs [0][j] = 0;

       for ( int i =1; i<=n; i++)

       {
          for ( int j =1; j<=m; j++)

          {
             if ( s1 [i-1] == s2[j-1])
             {
                lcs [i][j] = lcs[i-1] [j-1] + 1;
                dir [i] [j]= 'D';
             }
```

```
        else

        {
           if( lcs [i-1] [j] >= lcs [i][j-1] )

           {
              lcs [i][j] = lcs [i-1][j];
              dir [i] [j] = 'U';
           }

           else

           {
             lcs [i][j] = lcs [i][j-1];
              dir [i] [j] = 'L';
           }
        }
     }
  }

cout << " Your lcs table looks like this  " << endl;

 for ( int i=0; i<=n; i++)

 {
    for (int j =0; j<=m; j++)

    {
       cout<< lcs [i] [j] << "   ";
    }

    cout<< endl;
 }


  cout <<  endl << " The length of the lcs is : " << lcs [n] [m] << endl << endl;
```

```
  int i=n,j=m;
  string ans;

  while( i>0 && j>0)

  {
     if ( dir [i] [j] == 'U')

     {
        i--;
     }

     else  if ( dir [i] [j] == 'L')

     {
        j--;
     }

    else if ( dir [i] [j] == 'D')

    {
       ans+= s1[i-1];

       i--;

       j--;
    }

  }
  reverse ( ans.begin(),ans.end() );

  cout <<  endl << "  Your subsequence is  "  << ans ;

}
```

```
 Enter your strings
YBDCABA
XABCBDAB

 your LCS table looks like this
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  1  1  1  1  1  1
0  0  0  1  1  1  2  2  2
0  0  0  1  2  2  2  2  2
0  0  1  1  2  2  2  3  3
0  0  1  2  2  3  3  3  4
0  0  1  2  2  3  3  4  4

 length of LCS is 4

 your subsequence is BDAB
```

0/1 knapsack

```
#include<bits/stdc++.h>
using namespace std;

int main ()
{
   int no_it;

   cout<<"enter the number of items:  "  ;

   cin>>no_it;

   int weight[no_it],price[no_it];

   cout<< endl;

  cout << " enter weight accordingly " << endl;

   for (int i=0; i<no_it; i++)
   {
      cin>> weight[i] ;

   }

  cout << " enter price accordingly " << endl;

    for (int i=0; i<no_it; i++)
   {
      cin>> price[i] ;
   }

   int capacity;

   cout<<" select the knapsack capacity: ";

   cin>>capacity;
```

```
int knapsack[no_it+1][capacity+1];

for(int i=0; i<=no_it; i++) // rows will be 0 now, overall 1st column is 0 now
{
   knapsack[i][0]=0;
}
for (int j=0; j<=capacity; j++) // columns will be 0 now, overall 1st row is 0 now
{
   knapsack[0][j]=0;
}

for(int i=1; i<=no_it; i++)
{
   for (int j=1; j<=capacity; j++)
   {
        int w = weight[i-1];

      if(weight[i-1] <= j)
      {
         knapsack[i][j]=max(knapsack[i-1][j], knapsack[i-1] [j-w] +price[i-1]);
      }
      else
      {
         knapsack[i][j]=knapsack[i-1][j];
      }
   }
}

cout << endl<< " 0/1 knapsack table " << endl;

for(int i=0; i<=no_it; i++)
{
   for(int j=0; j<=capacity; j++)
   {
      cout<< knapsack[i][j]<<"  " ;
   }
   cout<<endl;
}

cout <<  " maximum profit  is "  <<  knapsack[no_it][capacity] <<endl;
```

```
  int i = no_it;

  int j = capacity;

  while ( i>0 && j>0 )

  {
     if ( knapsack[i][j] != knapsack[i-1][j])

     {

        cout << " item " << i << "  is Selected " << endl;

         j-= weight[i-1];

         i--;

     }

     else if (  knapsack[i][j] == knapsack[i-1][j ] )
     {

         cout << " item " <<   i  << "  is Not selected " << endl;

         i--;

     }
  }


}
```

```
 Enter weight accordingly
1
3
4
5

 Enter  price  accordingly
1
4
5
7

 Enter your knapsack capacity 7

 Your 0/1 knapsack table is :
0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1
0 1 1 4 5 5 5 5
0 1 1 4 5 6 6 9
0 1 1 4 5 7 8 9

 Maximum profit is 9
 item 4 is not selected
 item 3 is selected
 item 2 is selected
```

Minimum number of coins

```
#include<bits/stdc++.h>
using namespace std;

int main ()
{
   int n,m;

   cout<<" enter the number of coins:  "  ;

   cin>> n;

   int coins[n];

   cout <<  endl << " Enter the amount you want to generate:  ";

   cin>> m;

   int dp[n+1][m+1];

   cout<< endl;

  cout << " enter coins accordingly " << endl;

   for (int i=0; i<n; i++)

   {
      cin>> coins[i] ;
   }

   for(int i=0; i<=n; i++)
   {
     dp [i][0]=0;
   }
   for (int j=1; j<=m; j++)
   {
      dp[0][j]=m+1;
   }
```

```
   for(int i=1; i<=n; i++)
   {
      for (int j=1; j<=m; j++)
      {
         int c=coins[i-1];

         if( j >= coins[i-1])
         {
            dp[i][j]=min(dp[i-1][j], 1+ dp [i] [j - c]);
         }
         else
         {
            dp[i][j]= dp [i-1][j];
         }

      }
   }

   cout <<  endl << " coin change table " << endl << endl;

   for(int i=0; i<=n; i++)
   {
      for(int j=0; j<=m; j++)
      {
         cout<< dp[i][j]<<" " ;
      }
      cout<<endl;
   }


   cout <<  endl << " minimum number of coins needed here is  "  <<  dp[n][m] <<endl;


   int i = n;

   int j = m;

     //printf(" \n  before entering into the while loop, value of  n = %d and m = %d  \n\n ",
n,m );
```

```
   while ( i>0 && j>0 )

   {
      if ( dp[i][j] == dp[i-1][j])

      {
          cout << " coin " <<   coins[i-1]  << "  is Not selected " << endl;

         i--;

         //cout<<  " value of i  = " << i << " value of j  = " <<  j <<endl;

      }

      else if (  dp[i][j] != dp [i-1][j] )
      {


        cout << " coin  " <<  coins[i-1] << "  is Selected " << endl;

         j-=coins[i-1];

          //cout<<  " value of i  = " << i << " value of j  = " <<  j <<endl;

      }
   }

}
```

```
 Enter the number of coins :  4

 Enter the amount you want to generate : 10

 Enter coins accordingly
1
5
6
9
  your minimum number of coin change table is
0 11 11 11 11 11 11 11 11 11 11
0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 1 2 3 4 5 2
0 1 2 3 4 1 1 2 3 4 2
0 1 2 3 4 1 1 2 3 1 2

 Minimum number of coins  needed here is 2

 coin 9 is  not selected
 coin 6 is  not selected
 coin 5 is selected
 coin 5 is selected
```

Maximum number of ways

```
#include<bits/stdc++.h>
using namespace std;

int main ()

  {

   int n,m;

   cout<<" enter the number of coins:  "  ;

   cin>> n;

   int coins[n];

   cout << endl << " Enter the amount you want to generate: ";

   cin>> m;

   int dp[n+1][m+1];

   cout<< endl;

    cout << " enter coins accordingly " << endl;

   for (int i=0; i<n; i++)
   {
      cin>> coins[i] ;
   }
```

```
for(int i=0;i<=n;i++)
   {
      dp[i][0]=1;
   }

    for(int j=1;j<=m;j++)
   {
      dp[0][j]=0;
   }

   for(int i=1;i<=n;i++)
   {

      for(int j=1;j<=m;j++)
      {
          int c = coins[i-1];

         if ( j>= coins [i-1] )

         {
             dp[i][j]=(dp[i-1][j] + dp[i] [j - c]);
         }

         else

         {
             dp[i][j]=dp[i-1][j];
         }

      }


   }

     cout <<  endl << " dp table of the maximum number of ways " << endl << endl ;
```

```
 for(int i=0; i<=n; i++)
  {
     for(int j=0; j<=m; j++)
     {
        cout<< dp[i][j]<<" " ;
     }

     cout<<endl;
  }


   cout <<  endl << "The maximum number of ways:   " <<  dp[n][m] << endl ;

    //cout<<" The value of n is " << n <<  "   " <<  "The value of m is "<< m <<
endl;

  }
```

```
 enter the number of coins:  3

 Enter the amount you want to generate:  5

 enter coins accordingly
1
2
5

 dp table of the maximum number of ways

1 0 0 0 0 0
1 1 1 1 1 1
1 1 2 2 3 3
1 1 2 2 3 4

 Maximum number of ways is 4
```

```
 enter the number of coins:  3

 Enter the amount you want to generate:  4

 enter coins accordingly

1
2
5

 dp table of the maximum number of ways

1 0 0 0 0
1 1 1 1 1
1 1 2 2 3
1 1 2 2 3

 Maximum number of ways is 3
```

```
 enter the number of coins: 3

 Enter the amount you want to generate: 6

 enter coins accordingly
1
3
5

 dp table of the maximum number of ways

1 0 0 0 0 0 0
1 1 1 1 1 1 1
1 1 1 2 2 2 3
1 1 1 2 2 3 4

 Maximum number of ways is 4
```