Practical No. 04 Write a C/C++ code for the following Algorithm with an explanation.

1. Travelling salesman Problem.

#include <stdio.h>

*int* matrix[25][25], visited\_cities[10], limit, cost = 0;

*int* tsp(*int* *c*)

{

*int* count, nearest\_city = 999;

*int* minimum = 999, temp;

 for(count = 0; count < limit; count++)

 {

 if((matrix[*c*][count] != 0) && (visited\_cities[count] == 0))

 {

 if(matrix[*c*][count] < minimum)

 {

 minimum = matrix[count][0] + matrix[*c*][count];

 }

 temp = matrix[*c*][count];

 nearest\_city = count;

 }

 }

 if(minimum != 999)

 {

 cost = cost + temp;

 }

 return nearest\_city;

}

*void* minimum\_cost(*int* *city*)

{

*int* nearest\_city;

 visited\_cities[*city*] = 1;

 printf("%d ", *city* + 1);

 nearest\_city = tsp(*city*);

 if(nearest\_city == 999)

 {

 nearest\_city = 0;

 printf("%d", nearest\_city + 1);

 cost = cost + matrix[*city*][nearest\_city];

 return;

 }

 minimum\_cost(nearest\_city);

}

*int* main()

{

*int* i, j;

 printf("Enter Total Number of Cities:\t");

 scanf("%d", &limit);

 printf("\nEnter Cost Matrix\n");

 for(i = 0; i < limit; i++)

 {

 printf("\nEnter %d Elements in Row[%d]\n", limit, i + 1);

 for(j = 0; j < limit; j++)

 {

 scanf("%d", &matrix[i][j]);

 }

 visited\_cities[i] = 0;

 }

 printf("\nEntered Cost Matrix\n");

 for(i = 0; i < limit; i++)

 {

 printf("\n");

 for(j = 0; j < limit; j++)

 {

 printf("%d ", matrix[i][j]);

 }

 }

 printf("\n\nPath:\t");

 minimum\_cost(0);

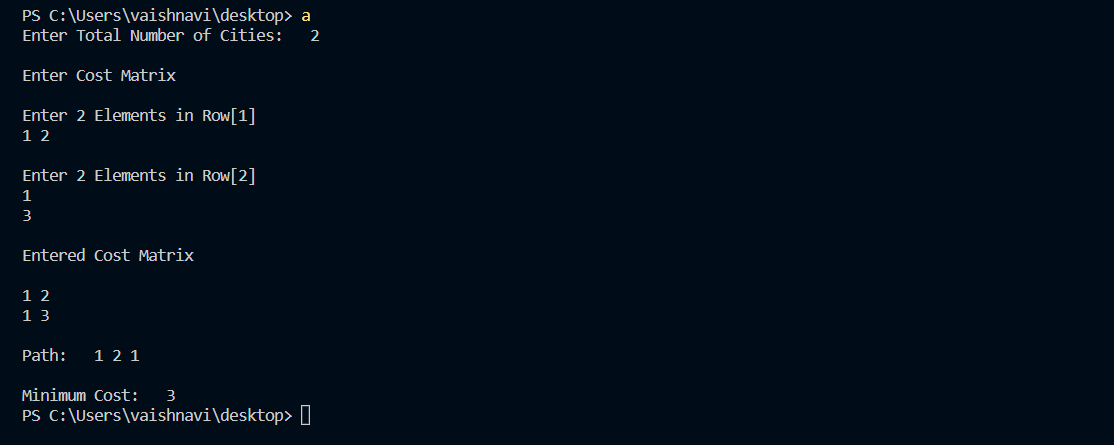
 printf("\n\nMinimum Cost: \t");

 printf("%d\n", cost);

 return 0;

}

Output:



1. BF string Matching Algorithm

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

*void* main()

{

*int* i,j,k,n,m,flag=0;

*char* t[40],p[30];

 printf("Enter text: ");

 gets(t);

 printf("\nEnter pattern: ");

 gets(p);

 n=strlen(t);

 m=strlen(p);

 for(i=0;i<=n-m;i++)

 {

  j=0;

  while(j<m && p[j]==t[j+i])

   {

     j++;

     if(j==m)

     {

      flag=1;

      k=i+1;

     }

     else

      flag=0;

   }

 }

 if(flag==1)

    printf("\nPattern found at position: %d\n ",k);

 else

    printf("\nPattern not found in text \n");

  getch();

}

Output:

