**Axborot tizimlari va texnologiylari yo’nalishi**

**122-20-guruh talabasi**

**Sattorova Mohiraning**

**Algoritmlar va berilganlar strukturasi fanidan**

1. **topshirig’I**

**9-Topshiriq:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace struktura\_9

{

class Program

{

public static readonly int M = 999999; // erishib bo'lmaydigan masofani ko'rsatadi

static int[,] map = new int[,] {

{ 0,79,77,35,64,92,50},

{ 79,0,43,27,57,69,94},

{ 77,43,0,87,71,65,56},

{ 35,27,87,0,43,49,30},

{ 64,57,71,43,0,74,91},

{ 92,69,65,49,74,0,9},

{50,94,56,30,91,9,0}

};

public static readonly int N = (int)Math.Sqrt(map.Length);

static bool[] visit = new bool[N];

static int[] closest = new int[N];

static int[] lowcost = new int[N];

static readonly int start = 0;

static void Main(string[] args)

{

kruska();

string str1 = "Eng yaqin qiymat:";

string str2 = "Arzon narxdagi qiymat:";

int sum = 0;

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

{

str1 += closest[i] + " ";

str2 += lowcost[i] + " ";

sum += lowcost[i];

}

Console.WriteLine(str1);

Console.WriteLine(str2);

Console.WriteLine("Yig'indi:" + sum);

Console.ReadKey();

}

static void kruska()

{

visit[start] = true;

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

{

if (i != start)

{

lowcost[i] = map[start, i]; // Og'irliklarni boshqa qirralardan boshlang

closest[i] = start; // Eng yaqin o'zgarishning boshlanish nuqtasini belgilang

visit[i] = false; // Boshlashdan boshqa elementlarni ishga tushirish U to'plamiga tegishli emas

}

else

lowcost[i] = 0;

}

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

{

int temp = M;

int t = start;

for (int j = 0; j < N; j++) // V-U to'plamida U to'plamiga eng yaqin t cho'qqisini toping

{

if ((!visit[j]) && (lowcost[j] < temp))

{

t = j;

temp = lowcost[j];

}

}

if (t == start)

break;

visit[t] = true;

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

{

if ((!visit[j]) && (map[t, j] < lowcost[j]))

{

lowcost[j] = map[t, j];

closest[j] = t;

}

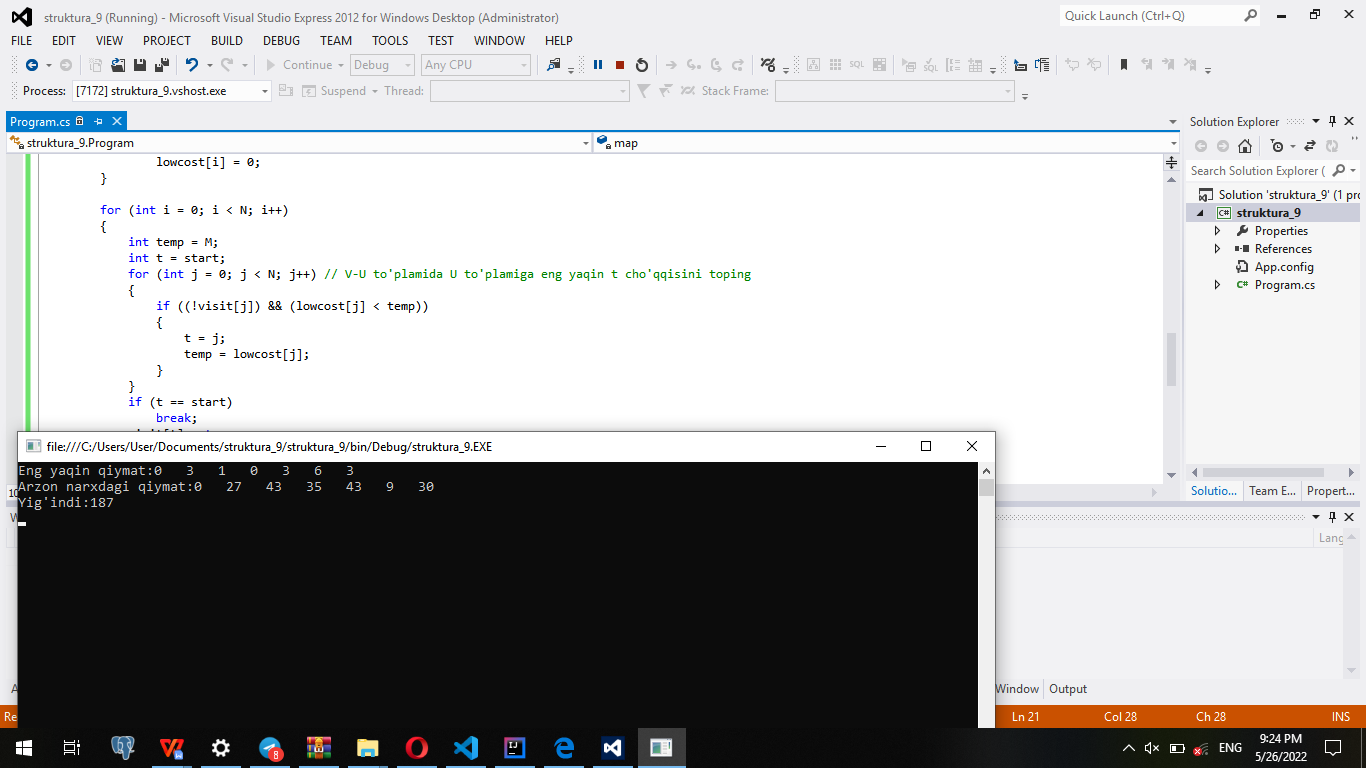
}

}

}

}

}



**9-topshiriq**

import sys  
  
def to\_be\_visited():  
 global visited\_and\_distance  
 v = -10  
 for index in range(number\_of\_vertices):  
 if visited\_and\_distance[index][0] == 0 \  
 and (v < 0 or visited\_and\_distance[index][1] <= \  
 visited\_and\_distance[v][1]):  
 v = index  
 return v  
  
vertices=[  
 [0,1,1,1,1,1],  
 [1,0,1,1,1,1],  
 [1,1,0,1,1,1],  
 [1,1,1,0,1,1],  
 [1,1,1,1,0,1],  
 [1,1,1,1,1,0]  
 ]  
edges=[  
 [0, 79, 77, 35, 64, 92, 50],  
 [79, 0, 43, 27,57, 69, 94],  
 [77, 43, 0, 87, 71, 65, 56],  
 [35, 27, 87, 0, 43, 49, 30],  
 [64, 57, 71, 43, 0, 74, 91],  
 [92, 69, 65, 49, 79, 0, 9],  
 [50, 94, 56, 30, 91, 59, 0]  
 ]  
number\_of\_vertices = len(vertices[0])  
visited\_and\_distance = [[0, 0]]  
for i in range(number\_of\_vertices-1):  
 visited\_and\_distance.append([0, sys.maxsize])  
  
for vertex in range(number\_of\_vertices):  
   
 to\_visit = to\_be\_visited()  
 for neighbor\_index in range(number\_of\_vertices):  
   
 if vertices[to\_visit][neighbor\_index] == 1 and \  
 visited\_and\_distance[neighbor\_index][0] == 0:  
 new\_distance = visited\_and\_distance[to\_visit][1] \  
 + edges[to\_visit][neighbor\_index]  
 if visited\_and\_distance[neighbor\_index][1] > new\_distance:  
 visited\_and\_distance[neighbor\_index][1] = new\_distance  
 visited\_and\_distance[to\_visit][0] = 1  
  
i = 0   
  
   
for distance in visited\_and\_distance:  
 print(chr(ord('a') + i)," dan boshlanuvchi qisqa yo'l uzunligi ",distance[1]," ga teng")  
 i = i + 1

