|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Artificial Intelligence  Practical #7 | | |
|  |  |  |  |
| **Name** | Sandeep Jain | **Roll Number** | 21302C0058 |
| **Subject/Course:** | **Artificial Intelligence** | | |
| **Topic** |  | | |
|  | | | |
| **1. Write a program to shuffle Deck of cards.** | | | |
| **Code: in visual studio 2017**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace ConsoleApp5  {  class Program  {  public static void shuffle(int []a)  {  Random rand = new Random();  for(int i=0;i<=51;i++)  {  int r = rand.Next(52); //0-51  int temp = a[i];  a[i] = a[r];  a[r] = temp;  }  }  static void Main(string[] args)  {  int[] a = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51 };  shuffle(a);  foreach(int x in a)  {  Console.Write(x + " ");    }  Console.ReadLine();  }  }  }  **Output in visual studio 2017** | | | |
| **2.Solve traveling salesman problem using artificial intelligence technique.** | | | |
| **Code: in codeblock**  #include<bits/stdc++.h>  using namespace std;  #define V 4  int travllingSalesmanProblem (int graph[][V], int s)  {  vector < int >vertex;  for (int i = 0; i < V; i++)  if (i != s)  vertex.push\_back (i);  int min\_path = INT\_MAX;  do  {  int current\_pathweight = 0;  int k = s;  for (int i = 0; i < vertex.size (); i++)  {  current\_pathweight += graph[k][vertex[i]];  k = vertex[i];  }  current\_pathweight += graph[k][s];  min\_path = min (min\_path, current\_pathweight);  }  while (next\_permutation (vertex.begin (), vertex.end ()));  return min\_path;  }  int main ()  {  int graph[][V] = { {0, 10, 15, 20},  {10, 0, 35, 25},  {15, 35, 0, 30},  {20, 25, 30, 0}  };  int s = 0;  cout <<”lowest path cost is: "<< travllingSalesmanProblem (graph, s) << endl;  return 0;  }  **Output:in codeblock**    **Code in python**  **from sys import maxsize**  **from itertools import permutations**  **V = 4**  **def travllingSalesmanProblem(graph, s):**    **vertex = []**  **for i in range(V):**  **if i != s:**  **vertex.append(i)**  **min\_path = maxsize**  **next\_permutation = permutations(vertex)**  **for i in next\_permutation:**  **current\_pathweight = 0**  **k = s**  **for j in i:**  **current\_pathweight += graph[k][j]**  **k = j**  **current\_pathweight += graph[k][s]**  **min\_path = min(min\_path, current\_pathweight)**  **return ("the cost is " , min\_path)**  **if \_\_name\_\_ == "\_\_main\_\_":**  **graph = [[0, 10, 15, 20], [10, 0, 35, 25], [15, 35, 0, 30], [20, 25, 30, 0]]**  **s = 0**  **print(travllingSalesmanProblem(graph, s))**  **Output** | | | |
|  | | | |
|  | | | |