

**Panipat Institute of Engineering & Technology,
Samalkha (Haryana)
Computer Science & Engineering Department**



Design and Analysis of algorithms Lab

PC-CS214AL

Submitted to:

Shivani Gaba
(Assistant Professor CSE)

Submitted by:

Anmol Baranwal

2820208

B.Tech CSE 4th Sem A3

Affiliated to



Kurukshetra University Kurukshetra, India

S no.	Name of the Program	Page No	Date
1	Write a program to implement Linear Search		
2	Use divide and conquer method to recursively implement Binary Search		
3	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.		
4	Using Open, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.		
5	Write a program to implement Longest Common Subsequence		
6	Write a program to implement Matrix Chain Multiplication		
7.a.	Obtain the Topological ordering of vertices in a given digraph.		
7.b.	Compute the transitive closure of a given directed graph using Warshall's algorithm.		
8	Implement 0/1 Knapsack problem using Dynamic Programming.		
9	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.		
10	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.		
11.a.	Print all the nodes reachable from a given starting node in a digraph using BFS method		

Sno.	Name of the Program	Page No	Date
11.b.	Check whether a given graph is connected or not using DFS method		
12	Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.		
13	Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.		
14	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.		
15	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm. Parallelize this algorithm, implement it using Open and determine the speed-up achieved.		
16	Implement N Queen's problem using Back Tracking.		

Experiment 1.**Write a Program to implement LINEAR SEARCH****Program:-**

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

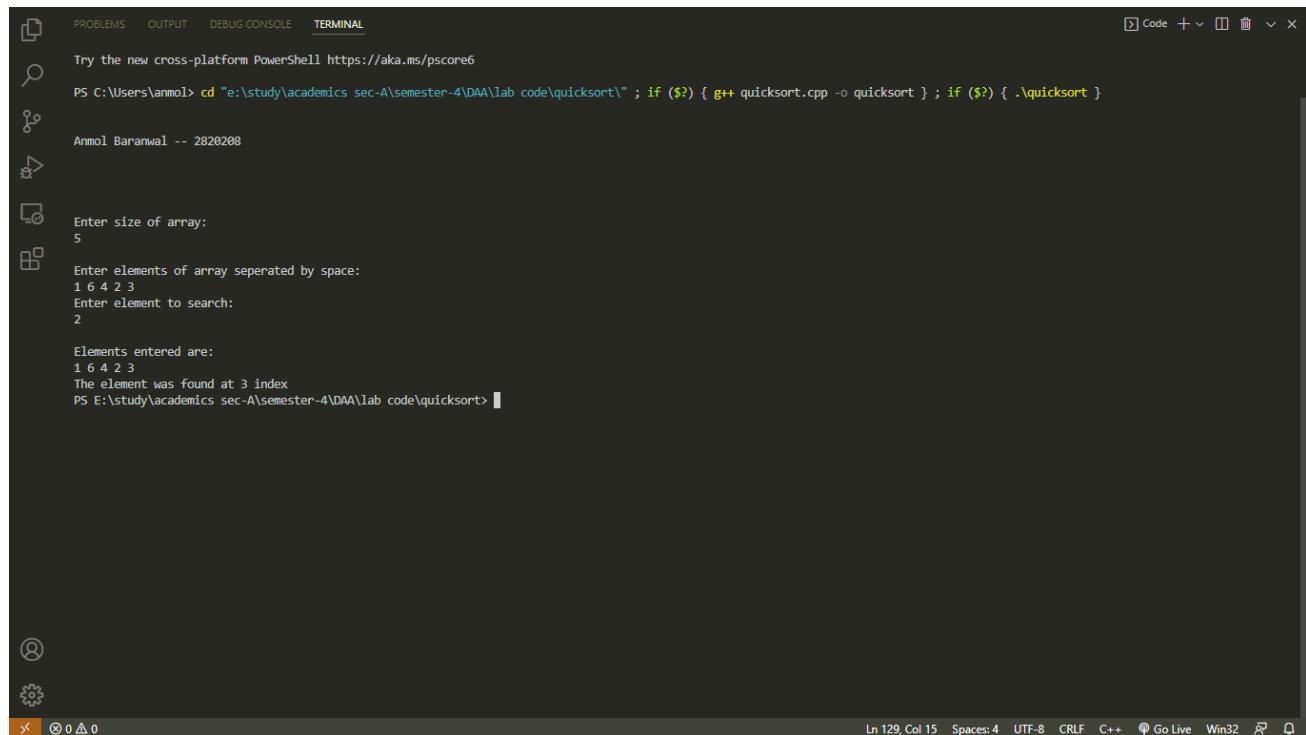
void displayArray(int arr[], int n)
{
    for(int i=0;i<n;i++){
        cout<<arr[i]<<" ";
    }
}

int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int n, arr[50];
    cout<<"\n\n";
    cout<<"Enter size of array: \n";
    cin>>n;
    cout<<"\nEnter elements of array seperated by space:\n";
    for(int i=0;i<n;i++){
        cin>> arr[i];
    }

    int key;
    cout<<"Enter element to search:\n";
    cin>>key;
    for(int i=0;i<n;i++){
        if(arr[i]==key){
            key = i;
            break;
        }
    }
    cout<<"\nElements entered are:\n";
    displayArray(arr,n);
    cout<<"\nThe element was found at "<< key <<" index";
}
```

```
    return 0;  
}
```

OUTPUT:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
PS C:\Users\anmol> cd "E:\study\academics sec-A\semester-4\DAA\lab code\quicksort" ; if ($?) { g++ quicksort.cpp -o quicksort } ; if ($?) { ./quicksort }  
Anmol Baranwal -- 2820208  
Enter size of array:  
5  
Enter elements of array seperated by space:  
1 6 4 2 3  
Enter element to search:  
2  
Elements entered are:  
1 6 4 2 3  
The element was found at 3 index  
PS E:\study\academics sec-A\semester-4\DAA\lab code\quicksort>
```

Experiment 2.

Use divide and conquer method to recursively implement Binary Search

Program:-

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

int binarySearch(int arr[], int start, int end, int key)
{
    while(start<=end){
        int mid = start + (end-start)/2;
        if(arr[mid]==key){
            return mid;
        } else if(arr[mid]>key){
            end=mid-1;
        } else{
            start=mid+1;
        }
    }
    return -1;
}

int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int n, arr[50];
    cout<<"\n\n";
    cout<<"Enter size of array: \n";
    cin>>n;
    cout<<"\nEnter elements of array seperated by space:\n";
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }

    int key;
```

```
cout<<"Enter element to search:\n";
cin>>key;

int result= binarySearch(arr, 0, n-1, key);

if(result==-1){
    cout<<"\nanswer is not found in array:";
} else {
    cout<<"answer is found at: "<<result+1;
}

cout<<"\n\n";

return 0;
}
```

OUTPUT:

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\quicksort\" ; if ($?) { g++ quicksort.cpp -o quicksort } ; if ($?) { ./quicksort }

Anmol Baranwal -- 2820208

Enter size of array:
6

Enter elements of array seperated by space:
1 2 6 9 11 14
Enter element to search:
11
answer is found at: 5

PS E:\study\academics sec-A\semester-4\DAA\lab code\quicksort>
```

The terminal window includes standard icons for file operations (New, Open, Save, etc.) and a status bar at the bottom with information like line and column counts, encoding, and file type.

Experiment 3.

Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Program:-

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

void swap(int* a,int* b){
    int temp=*a;
    *a=*b;
    *b=temp;
}

void generate_number(int a[], int n) {
    cout<<"The numbers are: \n";
    for (int i = 0; i < n; i++){
        a[i] = rand() % 1000;
        cout<<a[i]<<" ";
    }
    cout<<"\n";
}

void printArray(int arr[], int size)
{
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int partition(int A[], int l, int h)
{
    int pivot=A[l]; // pivot element
    int j=h+1,i=l;
    do{
```

```

do{i++;}while(A[i]<=pivot);
do{j--;}while(A[j]>pivot);           // finding greater element than pivot
                                         // finding smaller element than pivot

if(i<j){
    swap(&A[i],&A[j]);
}
}while(i<j);

swap(&A[l],&A[j]);

return j;
}

void Quicksort(int A[], int l, int h)
{
    if(l<h){
        int q=partition(A,l,h);
        Quicksort(A,l,q);    // calling left subarray after partition
        Quicksort(A,q+1,h);  // calling right subarray after partition
    }
}
}

int main() {

cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
int a[INT16_MAX], i, ch, n;

cout<<"Enter the number of the values to be generated\n";
cin>>i;
generate_number(a,i);
Quicksort(a, 0, i - 1);
cout<<"\nThe sorted array is: ";
for (n = 0; n < i; n++){
    cout<<a[n]<<" ";
}
return 0;
}

```

OUTPUT:

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter the number of the values to be generated
8
The numbers are:
41 467 334 500 169 724 478 358

The sorted array is: 41 169 334 358 467 478 500 724
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal window includes standard icons for file operations (New, Open, Save, etc.) and a status bar at the bottom showing line and column counts, encoding, and other settings.

Experiment 4.

Using Open, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Program:-

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

void Merge(int A[],int l,int mid,int h)
{
    int i=l,j=mid+1,k=l;
    int B[100];

    while(i<=mid && j<=h)
    {
        if(A[i]<A[j])
            B[k++]=A[i++];
        else
            B[k++]=A[j++];
    }
    for(;i<=mid;i++)
        B[k++]=A[i];
    for(;j<=h;j++)
        B[k++]=A[j];

    for(i=l;i<=h;i++)
        A[i]=B[i];
}

void recursiveMergeSort(int A[], int l, int h)
{
    int mid;
    if(l<h){
        mid=(l+h)/2;
        recursiveMergeSort(A,l,mid);
        recursiveMergeSort(A,mid+1,h);
```

```

        Merge(A,l,mid,h);
    }
}

void generate_number(int a[], int n) {

    cout<<"The numbers are: \n";
    for (int i = 0; i < n; i++){
        a[i] = rand() % 1000;
        cout<<a[i]<<" ";
    }
    cout<<"\n";
}

int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int a[INT16_MAX], i, ch, n;
    cout<<"Enter the number of the values to be generated\n";
    cin>>i;
    generate_number(a,i);
    recursiveMergeSort(a, 0, i - 1);
    cout<<"\nThe sorted array after recursive mergesort is: ";
    for (n = 0; n < i; n++){
        cout<<a[n]<<" ";
    }
    cout<<"\n";
    return 0;
}

```

OUTPUT:

The screenshot shows a terminal window in a dark-themed code editor interface. The terminal tab is active, displaying a Windows PowerShell session. The output shows the following sequence of events:

- Windows PowerShell
- Copyright (C) Microsoft Corporation. All rights reserved.
- Try the new cross-platform PowerShell <https://aka.ms/pscore6>
- PS C:\Users\anmol> cd "E:\study\academics\sec-A\semester-4\DAA\lab code\all programs\" ; if (\$?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if (\$?) { .\tempCodeRunnerFile }
- Anmol Baranwal -- 2820208
- Enter the number of the values to be generated
- 7
- The numbers are:
- 41 467 334 500 169 724 478
- The sorted array after recursive mergesort is: 41 169 334 467 478 500 724
- PS E:\study\academics\sec-A\semester-4\DAA\lab code\all programs>

The status bar at the bottom indicates the current line (Ln 104), column (Col 1), and selection count (1262 selected). It also shows file statistics (Spaces: 4, UTF-8, CRLF, C++) and build configurations (Go Live, Win32).

Experiment 5.**Write a Program to implement Longest Common Subsequence****Program:-**

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

int LCS(char A[], char B[], int i, int j){
    int m = strlen(A);
    int n = strlen(B);
    if(i==m || j==n)
        return 0;
    else if(A[i]==B[j])
        return 1+LCS(A, B, i+1,j+1);
    else
        return max(LCS(A, B, i+1, j), LCS(A, B, i, j+1));
}
int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    char X[] = "AGGTAB";
    char Y[] = "GXTXAYB";

    int i=0,j=0;
    cout<<"\nThe first string is: ";
    for(int i=0;X[i]!='\0';i++){
        cout<<X[i];
    }
    cout<<endl;
    cout<<"\nThe second string is: ";
    for(int i=0;Y[i]!='\0';i++){
        cout<<Y[i];
    }
    cout<<endl;
    cout<<"Length of LCS is "<< LCS( X, Y, i, j );

    return 0;
}
```

OUTPUT:

The screenshot shows a terminal window in Visual Studio Code. The terminal tab is selected at the top. The output shows the following:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\quicksort"
PS E:\study\academics sec-A\semester-4\DAA\lab code\quicksort> if ($?) { g++ quicksort.cpp -o quicksort } ; if ($?) { .\quicksort }

Anmol Baranwal -- 2820208

The first string is: AGGTAB
The second string is: GXTXAYB
Length of LCS is 4
PS E:\study\academics sec-A\semester-4\DAA\lab code\quicksort>
```

The terminal interface includes standard VS Code icons for file operations, search, and help. The status bar at the bottom right shows the current line (Ln 291), column (Col 33), and encoding (UTF-8). It also displays icons for Go Live, Win32, and other development settings.

Experiment 6.**Write a Program to implement Matrix Chain Multiplication****Program:-**

```
#include <bits/stdc++.h>
using namespace std;
int dp[100][100];

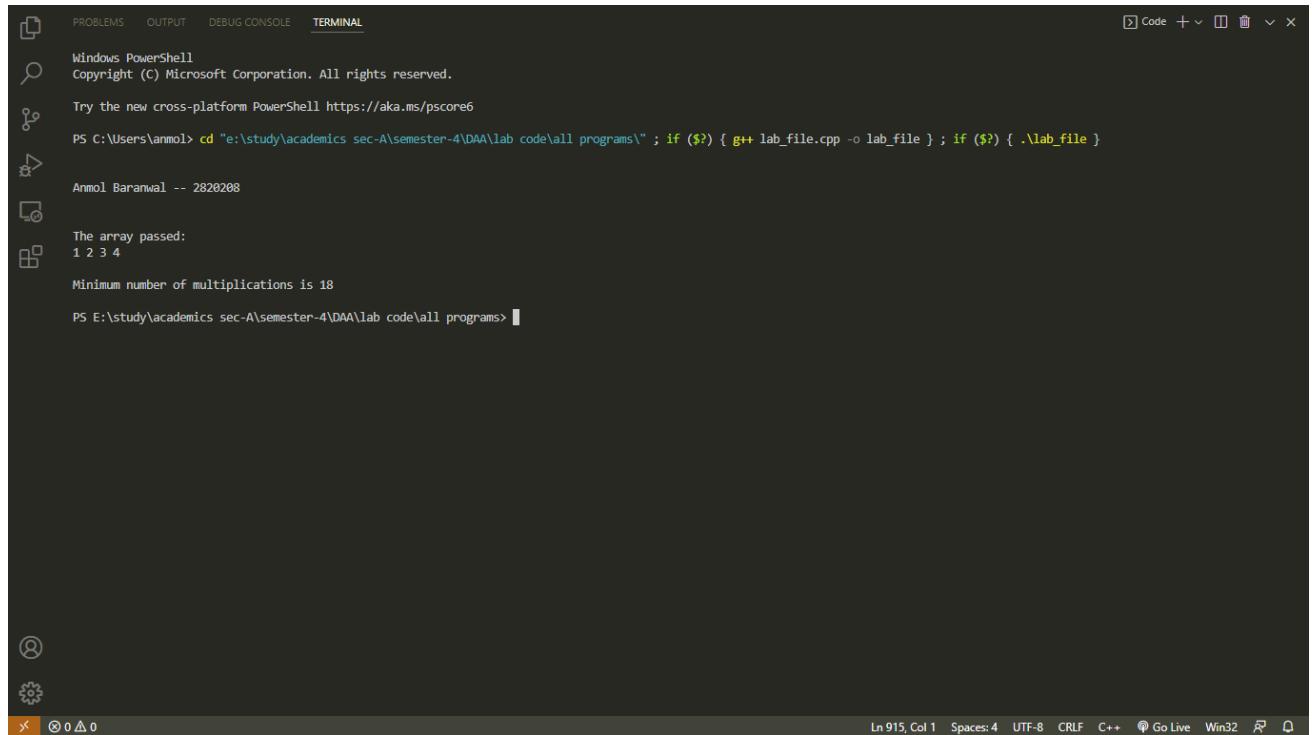
int matrixChainMemoised(int* p, int i, int j)
{
    if (i == j)
    {
        return 0;
    }
    if (dp[i][j] != -1)
    {
        return dp[i][j];
    }
    dp[i][j] = INT_MAX;
    for (int k = i; k < j; k++)
    {
        dp[i][j] = min(
            dp[i][j], matrixChainMemoised(p, i, k)
            + matrixChainMemoised(p, k + 1, j)
            + p[i - 1] * p[k] * p[j]);
    }
    return dp[i][j];
}
int MatrixChainOrder(int* p, int n)
{
    int i = 1, j = n - 1;
    return matrixChainMemoised(p, i, j);
}

int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int arr[] = { 1, 2, 3, 4 };
    int n = sizeof(arr) / sizeof(arr[0]);
    cout<<"\nThe array passed: \n";
}
```

```
for(int i=0;i<n;i++){
    cout<<arr[i]<<" ";
}
cout<<"\n\n";
memset(dp, -1, sizeof dp);

cout << "Minimum number of multiplications is "
<< MatrixChainOrder(arr, n);
cout<<"\n\n";
return 0;
}
```

OUTPUT:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

The array passed:
1 2 3 4

Minimum number of multiplications is 18

PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

Experiment 7.a.**Obtain the Topological ordering of vertices in a given digraph****Program:-**

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

int a[10][10], n, indegre[10];
void find_indegre() {
    int j, i, sum;
    for (j = 0; j < n; j++) {
        sum = 0;
        for (i = 0; i < n; i++)
            sum += a[i][j];
        indegre[j] = sum;
    }
}
void topology() {
    int i, u, v, t[10], s[10], top = -1, k = 0;
    find_indegre();
    for (i = 0; i < n; i++) {
        if (indegree[i] == 0) s[++top] = i;
    }
    while (top != -1) {
        u = s[top--];
        t[k++] = u;
        for (v = 0; v < n; v++) {
            if (a[u][v] == 1) {
                indegre[v]--;
                if (indegree[v] == 0) s[++top] = v;
            }
        }
    }
}
```

```
    }
}
cout<<"\n\nThe topological Sequence is:\n";
for (i = 0; i < n; i++)
    cout<<t[i]<<" ";
}
int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int i, j;
    cout<<"Enter number of jobs:";
    cin>>n;
    cout<<"\nEnter the adjacency matrix:\n";
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++)
            cin>>a[i][j];
    }
    topology();
    cout<<"\n";
    return 0;
}
```

OUTPUT:

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "E:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter number of jobs:6

Enter the adjacency matrix:
0 0 1 1 0 1
0 0 0 1 1 0
0 0 0 1 0 1
0 0 0 0 0 1
0 0 0 0 0 1
0 0 0 0 0 0

The topological Sequence is:
1 4 0 2 3 5
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal window has a dark background with light-colored text. It includes standard Windows PowerShell icons in the top-left corner and a toolbar with icons for Code, +, -, and other functions in the top-right corner. The bottom status bar shows file information like 'Ln 201, Col 24' and encoding settings like 'UTF-8'.

Experiment 7.b.

Compute the transitive closure of a given directed graph using Warshall's algorithm.

Program:-

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

int a[10][10], n;
void warshall() {
    int i, j, k;
    for (k = 1; k <= n; k++)
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n; j++)
                if (a[i][j] != 1) {
                    if (a[i][k] == 1 && a[k][j] == 1)
                        a[i][j] = 1;
                }
}
int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int i, j;
    cout << "Enter no of nodes : ";
    cin >> n;
    cout << "Enter adjacency matrix : \n";
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
            cin >> a[i][j];
    warshall();
    cout << "Transitive closure :\n";
    for (i = 1; i <= n; i++) {
```

```
for (j = 1; j <= n; j++)  
    cout << a[i][j] << "\t";  
    cout << endl;  
}  
return 0;  
}
```

OUTPUT:

The screenshot shows a terminal window with the following output:

```
Anmol Baranwal -- 2820208  
Enter no of nodes : 3  
Enter adjacency matrix :  
1 2 3  
4 5 6  
7 8 9  
Transitive closure :  
1 2 3  
4 5 6  
7 8 9  
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal window has a dark background and light-colored text. It includes standard terminal icons for file operations (copy, paste, delete) and navigation (back, forward). The status bar at the bottom shows the current line (Ln 258), column (Col 16), spaces (Spaces: 4), encoding (UTF-8), and file type (C++). It also includes icons for Go Live, Win32, and other development tools.

Experiment 8.**Implement 0/1 Knapsack problem using Dynamic Programming****Program:-**

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

int knapSackRec(int W, int wt[], int val[], int i, int** dp)
{
    if (i < 0)
        return 0;
    if (dp[i][W] != -1)
        return dp[i][W];

    if (wt[i] > W) {
        dp[i][W] = knapSackRec(W, wt, val, i - 1, dp);
        return dp[i][W];
    }
    else {
        dp[i][W] = max(val[i] + knapSackRec(W - wt[i], wt, val, i-1, dp),
                      knapSackRec(W, wt, val, i - 1, dp));
        return dp[i][W];
    }
}

int knapSack(int W, int wt[], int val[], int n)
{
    int** dp;
    dp = new int*[n];
    for (int i = 0; i < n; i++)
        dp[i] = new int[W + 1];
}
```

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < W + 1; j++)
        dp[i][j] = -1;
    return knapSackRec(W, wt, val, n - 1, dp);
}

int main()
{
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int n, W;
    cout<<"nEnter no of items: \n";
    cin>>n;
    cout<<"nEnter total weight of knapsack: \n";
    cin>>W;
    int val[n], wt[n];
    for(int i=0;i<n;i++){
        cout<<"nEnter the value and weight of item "<<i<<" seperated by space: ";
        cin>>val[i]>>wt[i];
    }
    cout <<"\nFinal value of knapsack is: "<< knapSack(W, wt, val, n);
    return 0;
}
```

OUTPUT:

The screenshot shows a terminal window in a dark-themed code editor interface. The terminal tab is active, displaying a Windows PowerShell session. The output shows the execution of a C++ program to solve a knapsack problem. The user inputs the number of items (3), the total weight capacity (50), and the value and weight pairs for three items. The program calculates and outputs the maximum value (220) that can be achieved without exceeding the weight limit.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter no of items:
3

Enter total weight of knapsack:
50

Enter the value and weight of item 0 seperated by space: 60 10
Enter the value and weight of item 1 seperated by space: 100 20
Enter the value and weight of item 2 seperated by space: 120 30

Final value of knapsack is: 220
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

Experiment 9.

From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm

Program:-

```
#include<bits/stdc++.h>
using namespace std;
const int inf = (int)1e9;

void dijkstra(int n, int source, int cost[10][10], int distance[]) {
    int i, v, u, visited[10], min, count = 2;
    for (i = 1; i <= n; i++)
        visited[i] = 0, distance[i] = cost[source][i];
    while (count <= n) {
        min = inf;
        for (u = 1; u <= n; u++)
            if (distance[u] < min && !visited[u])
                min = distance[u], v = u;
        visited[v] = 1;
        count++;
        for (u = 1; u <= n; u++)
            if ((distance[v] + cost[v][u] < distance[u]) && !visited[u])
                distance[u] = distance[v] + cost[v][u];
    }
}
int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int n, source, i, j, cost[10][10], distance[10];
    cout << "Enter the number of nodes : ";
    cin >> n;
```

```
cout << "Enter the cost matrix :\n";
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        cin >> cost[i][j];
        if (cost[i][j] == 0) cost[i][j] = inf;
    }
}

cout << "Enter the source : ";
cin >> source;
dijkstra(n, source, cost, distance);

cout << "Shortest path from given source are :\n";
for (i = 1; i <= n; i++)
    if (i != source)
        cout << source << " --> " << i << " : cost = " << distance[i] << endl;

return 0;
}
```

OUTPUT:

The screenshot shows a terminal window in a dark-themed code editor interface. The terminal tab is active, displaying a Windows PowerShell session. The command `cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\"` is run, followed by `if (\$?) { g++ lab_file.cpp -o lab_file } ; if (\$?) { .\lab_file }`. The user then enters their name and roll number: `Anmol Baranwal -- 2820208`. The program prompts for the number of nodes (5) and the cost matrix, which is a 5x5 grid of zeros and ones. It then asks for the source node (1) and prints the shortest paths from node 1 to all other nodes, along with their costs: 1 -> 2 : cost = 5, 1 -> 3 : cost = 12, 1 -> 4 : cost = 13, 1 -> 5 : cost = 12. The terminal window includes standard icons for file operations and a status bar at the bottom.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter the number of nodes : 5
Enter the cost matrix :
0 5 12 17 999
999 0 999 8 7
999 999 0 9 999
999 999 999 0 999
999 999 999 999 0
Enter the source : 1
Shortest path from given source are :
1 -> 2 : cost = 5
1 -> 3 : cost = 12
1 -> 4 : cost = 13
1 -> 5 : cost = 12
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

Experiment 10.**Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm****Program:-**

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

void kruskal(int cost[10][10], int n) {
    int parent[10], i, j, a, b, u, v, min, count = 1, sum = 0;
    for (i = 1; i <= n; i++)
        parent[i] = 0;

    while (count != n) {
        min = 9999;
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n; j++)
                if (cost[i][j] < min) {
                    min = cost[i][j];
                    u = a = i;
                    v = b = j;
                }
        while (parent[u] != 0)
            u = parent[u];
        while (parent[v] != 0)
            v = parent[v];
        if (u != v) {
            count++;
            sum = sum + cost[a][b];
            cout << "\nEdge (" << a << "," << b << ") : " << cost[a][b];
            parent[v] = u;
        }
        cost[a][b] = cost[b][a] = 9999;
    }
}
```

```
}

cout << "\n\nWeight of minimum spanning tree = " << sum << "\n\n";

}

int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int cost[10][10], i, j, n;

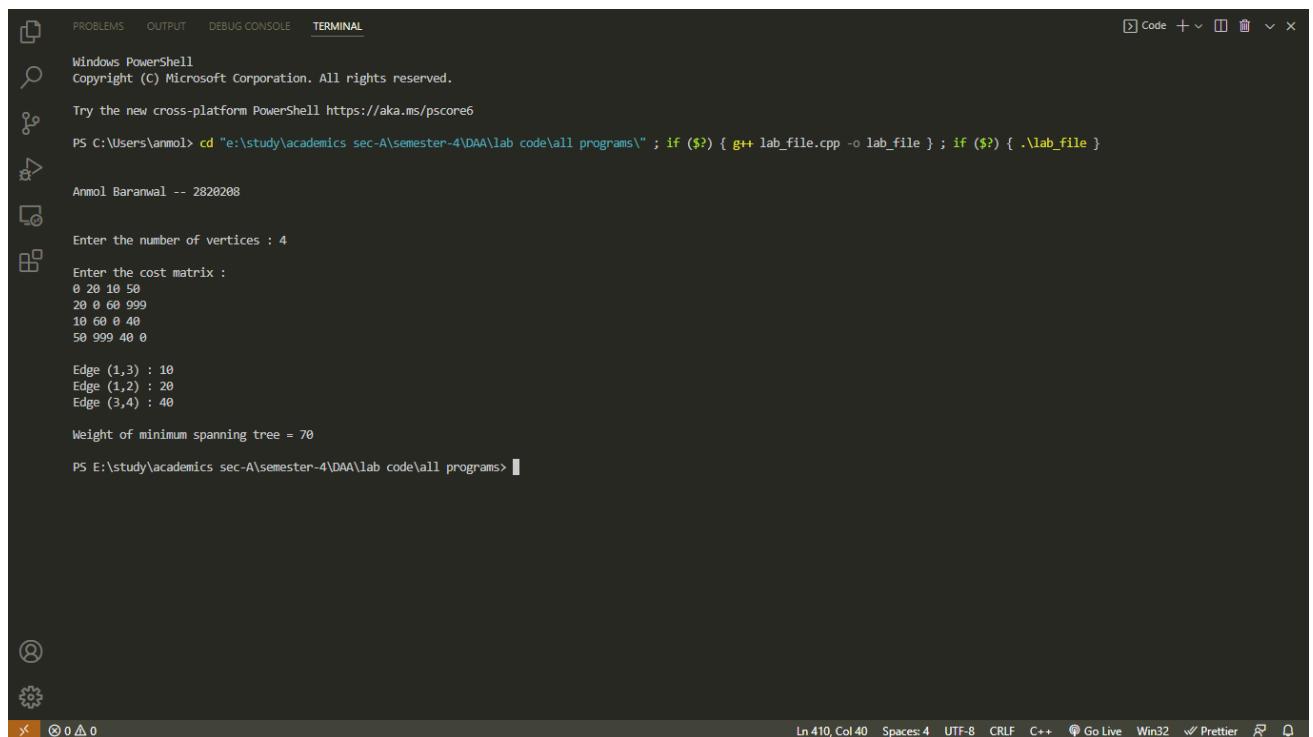
    cout << "\nEnter the number of vertices : ";
    cin >> n;

    cout << "\nEnter the cost matrix : \n";
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
            cin >> cost[i][j];

    kruskal(cost, n);

    return 0;
}
```

OUTPUT:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\anmol> cd "E:\study\academics sec-A\semester-4\DAA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter the number of vertices : 4
Enter the cost matrix :
0 20 10 50
20 0 60 999
10 60 0 40
50 999 40 0

Edge (1,3) : 10
Edge (1,2) : 20
Edge (3,4) : 40

Weight of minimum spanning tree = 70
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

Experiment 11.a.

Print all the nodes reachable from a given starting node in a digraph using BFS method

Program:-

```
#include<bits/stdc++.h>
using namespace std;
int visited[10];

void bfs(int n, int a[10][10], int source) {
    int i, q[10], u, front = 1, rear = 1;
    visited[source] = 1;
    q[rear] = source;
    while (front <= rear) {
        u = q[front];
        front++;
        for (i = 1; i <= n; i++)
            if (a[u][i] == 1 && visited[i] == 0) {
                rear++;
                q[rear] = i;
                visited[i] = 1;
            }
    }
}

int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int n, a[10][10], i, j, source;
    cout << "Enter the number of nodes : ";
    cin >> n;

    cout << "Enter the adjacency matrix :\n";
}
```

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= n; j++)
        cin >> a[i][j];

cout << "Enter the source : ";
cin >> source;

for (i = 1; i <= n; i++)
    visited[i] = 0;
bfs(n, a, source);

for (i = 1; i <= n; i++) {
    if (visited[i] == 0)
        cout << "The node " << i << " is NOT reachable.\n";
    else
        cout << "The node " << i << " is reachable.\n";
}
return 0;
}
```

OUTPUT:

The screenshot shows a terminal window titled "Windows PowerShell" running on a Windows operating system. The window title bar includes icons for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL being the active tab. The terminal content displays the following session:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter the number of nodes : 4

Enter the adjacency matrix :
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0

Enter the source : 1

The node 1 is reachable.
The node 2 is reachable.
The node 3 is reachable.
The node 4 is reachable.

PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal window has a dark background with light-colored text. The bottom status bar shows file statistics: Ln 459, Col 16, Spaces: 4, UTF-8, CRLF, C++, Go Live, Win32, Prettier, and a refresh icon.

Experiment 11.b.**Check whether a given graph is connected or not using DFS method****Program:-**

```
#include<bits/stdc++.h>
using namespace std;
int a[20][20], reach[20], n;
void dfs(int v) {
    int i;
    reach[v] = 1;
    for (i = 1; i <= n; i++)
        if (a[v][i] && !reach[i]) {
            cout<<"\n"<<v<<"->"<<i;
            dfs(i);
        }
}
int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int i, j, count = 0;
    cout<<"\nEnter number of vertices:";
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        reach[i] = 0;
        for (j = 1; j <= n; j++)
            a[i][j] = 0;
    }
    cout<<"\nEnter the adjacency matrix:\n";
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
            scanf("%d", &a[i][j]);
    dfs(1);
    printf("\n");
    for (i = 1; i <= n; i++) {
        if (reach[i])
            cout<<"1";
        else
            cout<<"0";
    }
}
```

```
    count++;
}
if (count == n)
    cout<<"\n Graph is connected\n";
else
    cout<<"\n Graph is not connected\n";
return 0;
}
```

OUTPUT:

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter number of vertices:4
Enter the adjacency matrix:
0 1 1 0
0 0 1 0
0 0 0 1
0 0 0 0

1->2
2->3
3->4

Graph is connected
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal shows the execution of a C++ program. It prompts for the number of vertices (4) and the adjacency matrix. The matrix is given as:

```
0 1 1 0
0 0 1 0
0 0 0 1
0 0 0 0
```

Then it lists the edges: 1->2, 2->3, 3->4. Finally, it outputs "Graph is connected".

Experiment 12.

Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.

Program:-

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

#include<bits/stdc++.h>
using namespace std;
#define MAX 20

int stk1[MAX];
int set1[MAX];
int size;
int top = -1;
int c = 1;
void push(int x) {
    if (top == MAX - 1) {
        cout << "No more place";
        exit(0);
    } else
        stk1[++top] = x;
}
void pop() {
    if (top < 0) {
        cout << "No elements";
        exit(0);
    } else
```

```

    top--;
}
void display() {
    cout << "Solution " << c << ":";
    c++;
    for (int i = 0; i <= top; i++)
        cout << stk1[i] << " ";
    cout << endl;
}
int subset(int pos, int sum) {
    int i;
    static int foundSoln = 0;
    if (sum > 0) {
        for (i = pos; i <= size; i++) {
            push(set1[i]);
            subset(i + 1, sum - set1[i]);
            pop();
        }
    }
    if (sum == 0) {
        display();
        foundSoln = 1;
    }
    return foundSoln;
}
int main() {
    cout << "\n\nAnmol Baranwal -- 2820208 \n\n";
    int i, sum;
    cout << "Enter the maximum number of elements:";
    cin >> size;
    cout << "Enter the elements:\n";
    for (i = 1; i <= size; i++)
        cin >> set1[i];
    cout << "Enter the required sum:";
    cin >> sum;
    cout << "\n";
    if (!subset(1, sum))
        cout << "No solution";
    return 0;
}

```

OUTPUT:

The screenshot shows a terminal window with the following content:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs\" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter the maximum number of elements:6
Enter the elements:
1 3 5 2 7 6
Enter the required sum:11

Solution 1:1 3 5 2
Solution 2:1 3 7
Solution 3:3 2 6
Solution 4:5 6

PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

The terminal window includes standard icons for file operations (New, Open, Save, etc.) and a status bar at the bottom with file information (Ln 531, Col 33), encoding (UTF-8), and other development tools like Go Live, Win32, Prettier, and a diff viewer.

Experiment 13.

Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

Program:-

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

int a[10][10], n, visit[10];
int cost_opt = 0, cost_apr = 0;
int least_apr(int c);
int least_opt(int c);

void mincost_opt(int city) {
    int i, ncity;
    visit[city] = 1;
    cout<<city<<"-->";
    ncity = least_opt(city);
    if (ncity == 999) {
        ncity = 1;
        cout<<ncity;
        cost_opt += a[city][ncity];
        return;
    }
    mincost_opt(ncity);
}

void mincost_apr(int city) {
    int i, ncity;
    visit[city] = 1;
    cout<<city<<"-->";
```

```

ncity = least_apr(city);
if (ncity == 999) {
    ncity = 1;
    cout<<ncity;
    cost_apr += a[city][ncity];
    return;
}
mincost_apr(ncity);
}

int least_opt(int c) {
    int i, nc = 999;
    int min = 999, kmin = 999;
    for (i = 1; i <= n; i++) {
        if ((a[c][i] != 0) && (visit[i] == 0))
            if (a[c][i] < min) {
                min = a[i][1] + a[c][i];
                kmin = a[c][i];
                nc = i;
            }
    }
    if (min != 999)
        cost_opt += kmin;
    return nc;
}

int least_apr(int c) {
    int i, nc = 999;
    int min = 999, kmin = 999;
    for (i = 1; i <= n; i++) {
        if ((a[c][i] != 0) && (visit[i] == 0))
            if (a[c][i] < kmin) {
                min = a[i][1] + a[c][i];
                kmin = a[c][i];
                nc = i;
            }
    }
    if (min != 999)
        cost_apr += kmin;
    return nc;
}
int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
}

```

```

int i, j;
cout<<"Enter No. of cities:\n";
cin>>n;

cout<<"Enter the cost matrix\n";
for (i = 1; i <= n; i++) {
    cout<<"Enter elements of row: "<<i<<"\n";
    for (j = 1; j <= n; j++)
        cin>>a[i][j];
    visit[i] = 0;
}
cout<<"The cost list is \n";
for (i = 1; i <= n; i++) {
    cout<<"\n\n";
    for (j = 1; j <= n; j++)
        cout<<"\t"<<a[i][j];
}
cout<<"\n\n Optimal Solution :\n";
cout<<"\n The path is :\n";
mincost_opt(1);
cout<<"\n Minimum cost:";
cout<<cost_opt;

cout<<"\n\n Approximated Solution :\n";
for (i = 1; i <= n; i++)
    visit[i] = 0;
cout<<"\n The path is :\n";
mincost_apr(1);
cout<<"\nMinimum cost:";
cout<<cost_apr;
cout<<"\n\nError in approximation is approximated solution/optimal solution=
"<<(float) cost_apr / cost_opt;

return 0;
}

```

OUTPUT:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\annol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter No. of cities:
4
Enter the cost matrix
Enter elements of row: 1
0 1 3 6
Enter elements of row: 2
1 0 2 3
Enter elements of row: 3
3 2 0 1
Enter elements of row: 4
6 3 1 0
The cost list is

      0      1      3      6
      1      0      2      3
      3      2      0      1
      6      3      1      0

Optimal Solution :

The path is :
1->2->4->3->1
Minimum cost:8

Approximated Solution :

The path is :
1->2->3->4->1
Minimum cost:10

Error in approximation is approximated solution/optimal solution= 1.25
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>

```

Ln 658, Col 93 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 Prettier

Experiment 14.

Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.

Program:-

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

int c[10][10], n;

void prims() {
    int visited[10], i, j, u, v, min, count = 1, sum = 0;
    for (i = 1; i <= n; i++)
        visited[i] = 0;
    visited[1] = 1;
    while (count != n) {
        min = 9999;
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n; j++)
                if (visited[i] == 1)
                    if (c[i][j] < min) {
                        min = c[i][j];
                        u = i;
                        v = j;
                    }
        if (visited[v] != 1) {
            cout << "\nEdge (" << u << "," << v << ") : " << min;
            visited[v] = 1;
            count++;
            sum = sum + min;
        }
        c[u][v] = c[v][u] = 9999;
    }
}
```

```
cout << "\nWeight of minimum spanning tree = " << sum << endl;
}

int main() {
    cout<<"\n\nAnmol Baranwal -- 2820208 \n\n";
    int i, j;

    cout << "Enter number of vertices : ";
    cin >> n;

    cout << "Enter cost matrix : \n";
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
            cin >> c[i][j];

    prims();

    return 0;
}
```

OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\anmol> cd "e:\study\academics sec-A\semester-4\DAA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

Enter number of vertices : 4
Enter cost matrix :
0 20 10 50
20 0 60 999
10 60 0 40
50 999 40 0

Weight of Edge (1,3) : 10
Weight of Edge (1,2) : 20
Weight of Edge (3,4) : 40
Weight of minimum spanning tree using prim's algorithm = 70
PS E:\study\academics sec-A\semester-4\DAA\lab code\all programs>
```

Experiment 15.

**Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
Parallelize this algorithm, implement it using Open and determine the
speed-up achieved.**

Program:-

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

Experiment 16.

Implement N Queen's problem using Back Tracking

Program:-

```
#include <bits/stdc++.h>
#define N 4
using namespace std;

void printSolution(int board[N][N])
{
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            cout << " " << board[i][j] << " ";
        printf("\n");
    }
}

bool isSafe(int board[N][N], int row, int col)
{
    int i, j;

    for (i = 0; i < col; i++)
        if (board[row][i])
            return false;

    for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
        if (board[i][j])
            return false;

    for (i = row, j = col; j >= 0 && i < N; i++, j--)
        if (board[i][j])
            return false;
}
```

```

return true;
}

bool solveNQUtil(int board[N][N], int col)
{
    if (col >= N)
        return true;

    for (int i = 0; i < N; i++) {
        if (isSafe(board, i, col)) {
            board[i][col] = 1;

            if (solveNQUtil(board, col + 1))
                return true;

            board[i][col] = 0; // BACKTRACK
        }
    }

    return false;
}

bool solveNQ()
{
    int board[N][N] = { { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 } };

    if (solveNQUtil(board, 0) == false) {
        cout << "Solution does not exist";
        return false;
    }

    printSolution(board);
    return true;
}

int main()
{
    cout << "\n\nAnmol Baranwal -- 2820208 \n\n";
    cout << "The values 1 indicate placement of queens: \n\n";
    solveNQ();
}

```

```
return 0;  
}
```

OUTPUT:

The screenshot shows a terminal window titled "Windows PowerShell" with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\anmol> cd "E:\study\academics sec-A\semester-4\DA\lab code\all programs" ; if ($?) { g++ lab_file.cpp -o lab_file } ; if ($?) { .\lab_file }

Anmol Baranwal -- 2820208

The values 1 indicate placement of queens:
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0

PS E:\study\academics sec-A\semester-4\DA\lab code\all programs>
```

The terminal shows the output of a C++ program that solves the N-Queens problem. It prints a 4x4 chessboard representation where '1' indicates the position of a queen and '0' indicates an empty square. The solution found is:

0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0