N-Queen Problem

```
#include<iostream>
using namespace std;

bool issafe(int **arr, int x,int y, int n){
    for (int row = 0; row < x; row++)
    {
        if (arr[row][y]==1)
        {
           return false;
        }

    }
    int row=x, col=y;

    while (row>=0 && col>=0)
    {
      if (arr[row][col]==1)
      {
        return false;
      }
      row--;
      col--;
    }
    row=x, col=y;
    while (row>=0 && col<n)
    {
      if (arr[row][col]==1)
      {
        return false;
      }
      row--;
      col++;
    }

    return true;

}

bool nqueen(int **arr, int x, int n){
    if (x>=n)
    {
        return true;
    }
    for (int col = 0; col < n; col++)
    {
        if (issafe(arr, x, col, n))
        {
            arr[x][col]=1;
```

```
            if (nqueen(arr, x+1, n))
            {
                return true;
            }
            arr[x][col]=0;

        }

    }
    return false;

}

int main(){
int n;
cout<<"Enter N * N Board Size: ";
cin>>n;
int **arr=new int*[n];
for (int i = 0; i < n; i++)
{
    arr[i]=new int[n];
    for (int j = 0; j < n; j++)
    {
        arr[i][j]=0;
    }

}

if (nqueen(arr,0,n))
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j <n; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

}


return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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

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

PS C:\Users\Asus\Desktop\DAA_Prac_02\N-Queen Problem> cd "c:\Users\A
cpp -o problem1 } ; if ($?) { .\problem1 }
Enter N * N Board Size: 4
Queen's Position:
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
PS C:\Users\Asus\Desktop\DAA_Prac_02\N-Queen Problem> cd "c:\Users\A
cpp -o problem1 } ; if ($?) { .\problem1 }
Enter N * N Board Size: 5
Queen's Position:
1 0 0 0 0
0 0 1 0 0
0 0 0 0 1
0 1 0 0 0
0 0 0 1 0
PS C:\Users\Asus\Desktop\DAA_Prac_02\N-Queen Problem>
```

Travelling Salesman Problem

```
#include <stdio.h>
int costmatrix[10][10], visited[10], n, cost = 0;

void start()
{
    int i, j;

    printf("Enter the number of villages: ");
    scanf("%d", &n);

    printf("\nEnter the Cost Matrix\n");

    for (i = 0; i < n; i++)
    {
        printf("Enter Elements of Row: %d\n", i + 1);

        for (j = 0; j < n; j++)
            scanf("%d", &costmatrix[i][j]);

        visited[i] = 0;
    }

    printf("\nThe cost list is:");

    for (i = 0; i < n; i++)
    {
        printf("\n");

        for (j = 0; j < n; j++)
            printf("\t%d", costmatrix[i][j]);
    }
}

void mincost(int city)
{
    int i, ncity;

    visited[city] = 1;

    printf("%d--->", city + 1);
    ncity = min(city);
```

```
    if (ncity == 999)
    {
        ncity = 0;
        printf("%d", ncity + 1);
        cost += costmatrix[city][ncity];

        return;
    }

    mincost(ncity);
}

int min(int c)
{
    int i, nc = 999;
    int min = 999, kmin;

    for (i = 0; i < n; i++)
    {
        if ((costmatrix[c][i] != 0) && (visited[i] == 0))
            if (costmatrix[c][i] + costmatrix[i][c] < min)
            {
                min = costmatrix[i][0] + costmatrix[c][i];
                kmin = costmatrix[c][i];
                nc = i;
            }
    }

    if (min != 999)
        cost += kmin;

    return nc;
}

int main()
{
    start();

    printf("\n\nThe Path is:\n");
    mincost(0);
    printf("\n\nMinimum cost is %d\n ", cost);

    return 0;
}
```

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\Asus\Desktop\DAA_Prac_02\Travelling Salesman Problem> cd "c:\Users\Asus\Desktop\DAA_Pra
 if ($?) { gcc TrueTSP.c -o TrueTSP } ; if ($?) { .\TrueTSP }
TrueTSP.c: In function 'mincost':
TrueTSP.c:42:13: warning: implicit declaration of function 'min' [-Wimplicit-function-declaration]
     ncity = min(city);
             ^~~
Enter the number of villages: 4

Enter the Cost Matrix
Enter Elements of Row: 1
0 4 1 3
Enter Elements of Row: 2
4 0 2 1
Enter Elements of Row: 3
1 2 0 5
Enter Elements of Row: 4
3 1 5 0

The cost list is:
        0       4       1       3
        4       0       2       1
        1       2       0       5
        3       1       5       0

The Path is:
1--->3--->2--->4--->1

Minimum cost is 7

PS C:\Users\Asus\Desktop\DAA_Prac_02\Travelling Salesman Problem>
```