INTERNAL ASSIGNMENT

INTERNAL ASSIGNMENT

Course Code: OMC-307 July 23

Course Title: Design and Analysis of Algorithms Lab

Last Date of Submission: 31/12/24

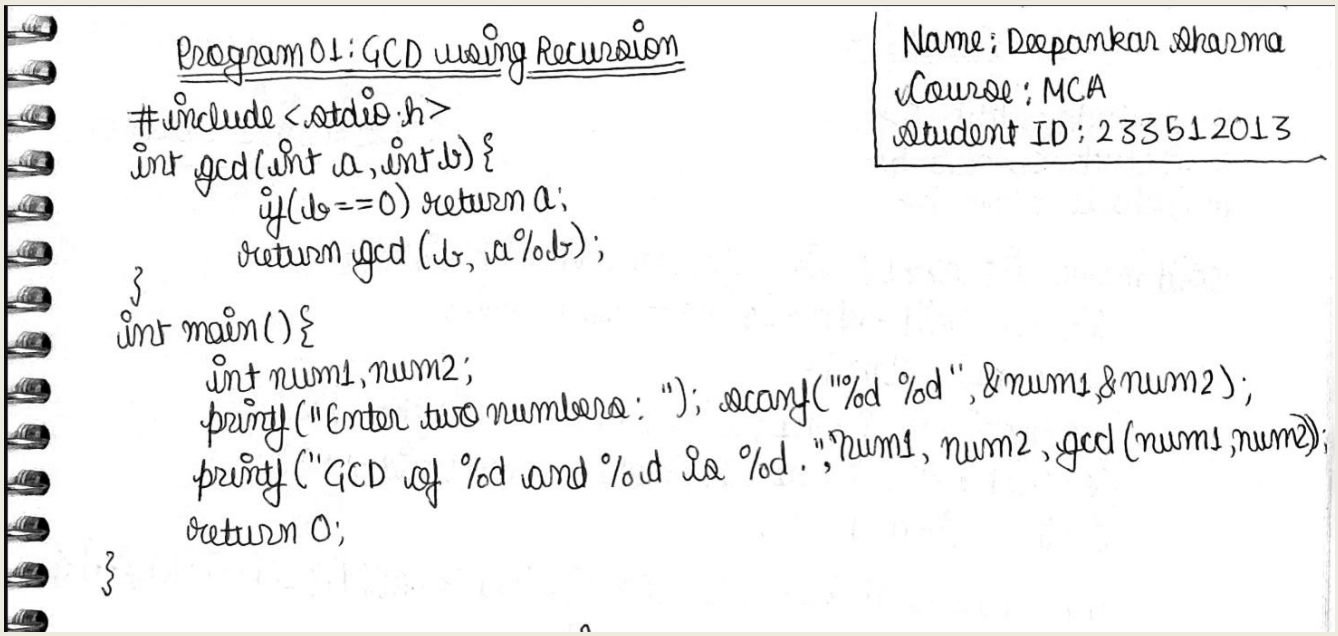
Maximum Marks: 30

Session: July 2024

Deepankar Sharma

233512013

1. Write a C Program using recursive function to find the GCD of two numbers.

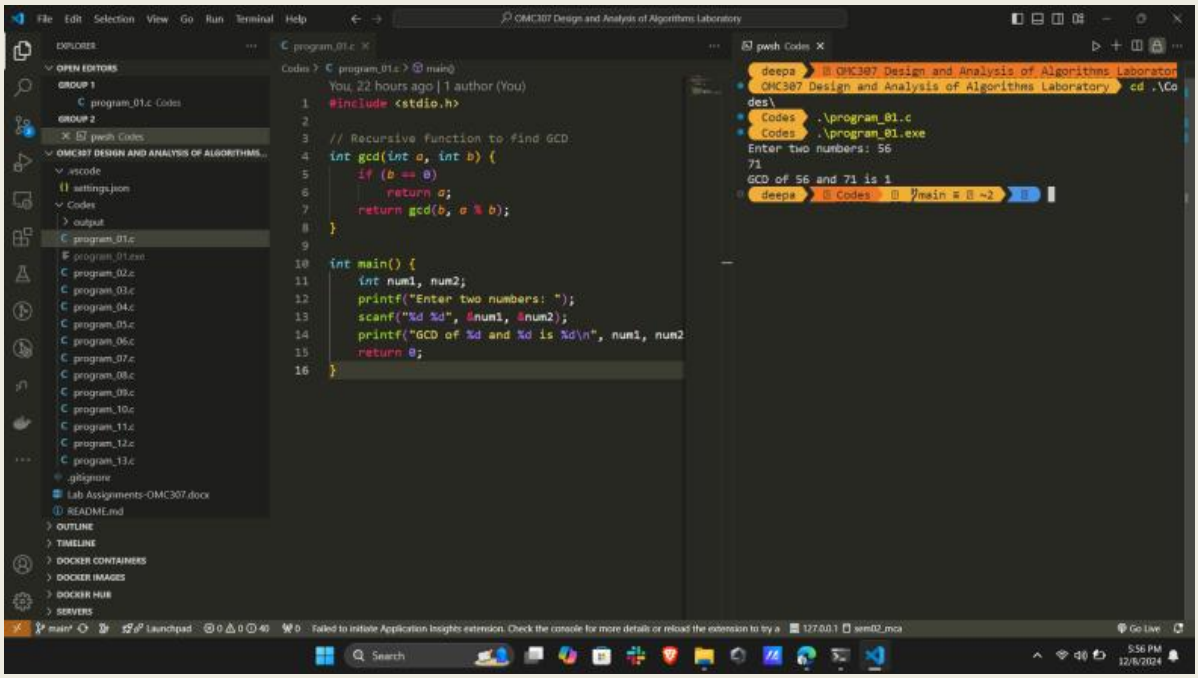


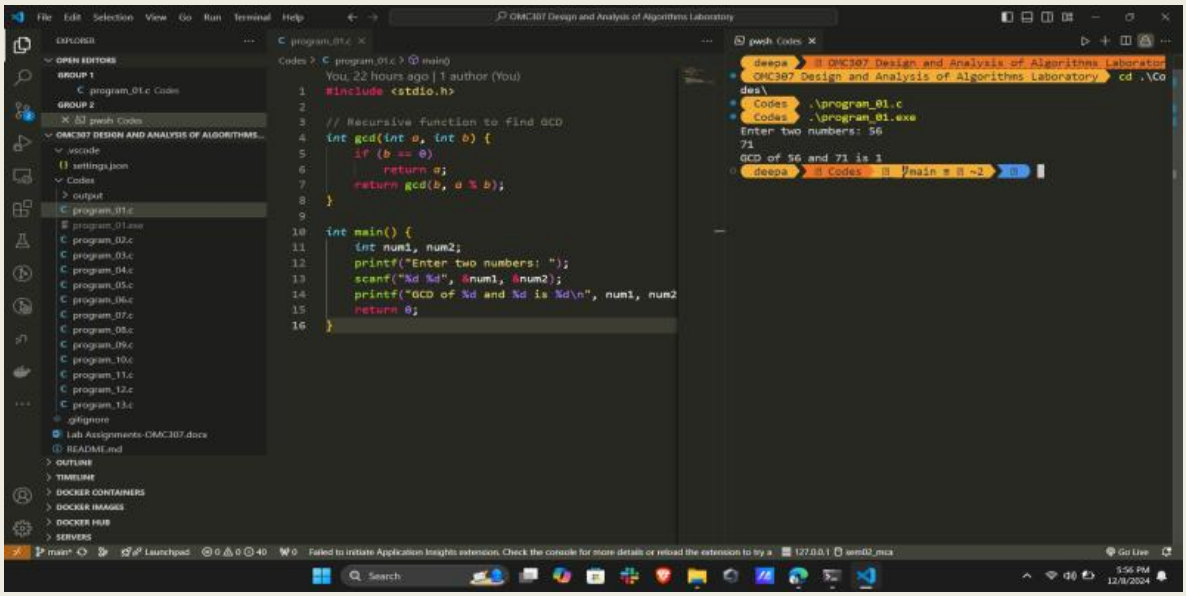
Program 01: GCD using Recursion

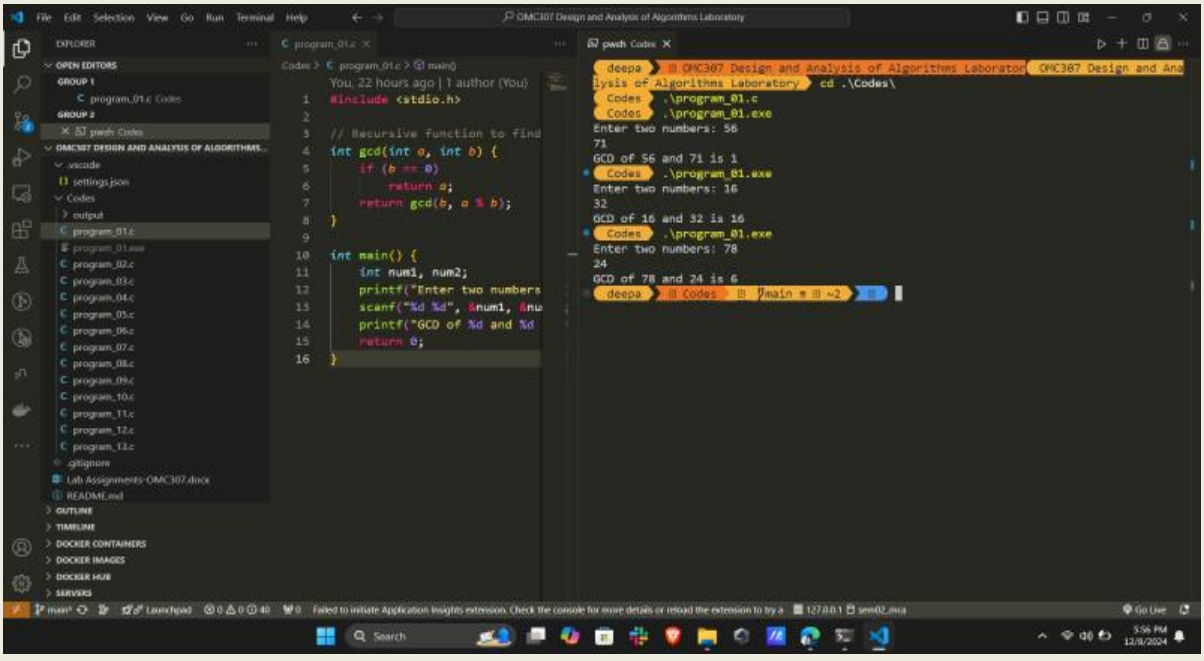
Name: Deepankar Sharma
Course: MCA
Student ID: 233512013

```
#include <stdio.h>
int gcd(int a, int b){
    if(b==0) return a;
    return gcd(b, a%b);
}
int main(){
    int num1, num2;
    printf("Enter two numbers: "); scanf("%d %d", &num1, &num2);
    printf("GCD of %d and %d is %d.", num1, num2, gcd(num1, num2));
    return 0;
}
```

Output


#include <stdio.h>
// Recursive function to find GCD
int gcd(int a, int b) {
if (b == 0)
return a;
return gcd(b, a % b);
}
int main() {
int num1, num2;
printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
printf("GCD of %d and %d is %d\n", num1, num2
return 0;
}
Enter two numbers: 56
71
GCD of 56 and 71 is 1


#include <stdio.h>
// Recursive function to find GCD
int gcd(int a, int b) {
if (b == 0)
return a;
return gcd(b, a % b);
}
int main() {
int num1, num2;
printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
printf("GCD of %d and %d is %d\n", num1, num2
return 0;
}
Enter two numbers: 56
71
GCD of 56 and 71 is 1

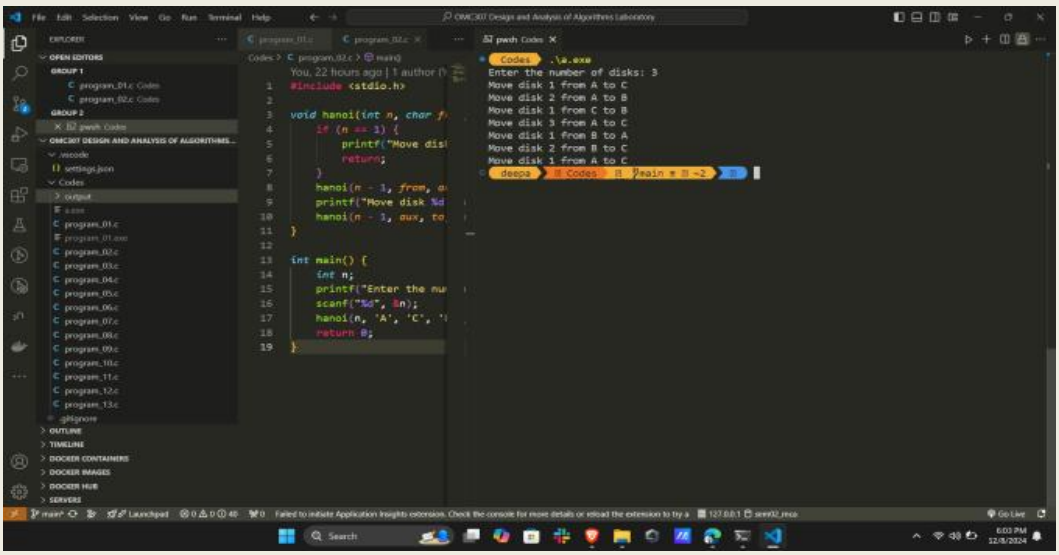

#include <stdio.h>
// Recursive function to find
int gcd(int a, int b) {
if (b == 0)
return a;
return gcd(b, a % b);
}
int main() {
int num1, num2;
printf("Enter two numbers
scanf("%d %d", &num1, &nu
printf("GCD of %d and %d
return 0;
}
Enter two numbers: 56
71
GCD of 56 and 71 is 1
Enter two numbers: 16
32
GCD of 16 and 32 is 16
Enter two numbers: 78
24
GCD of 78 and 24 is 6

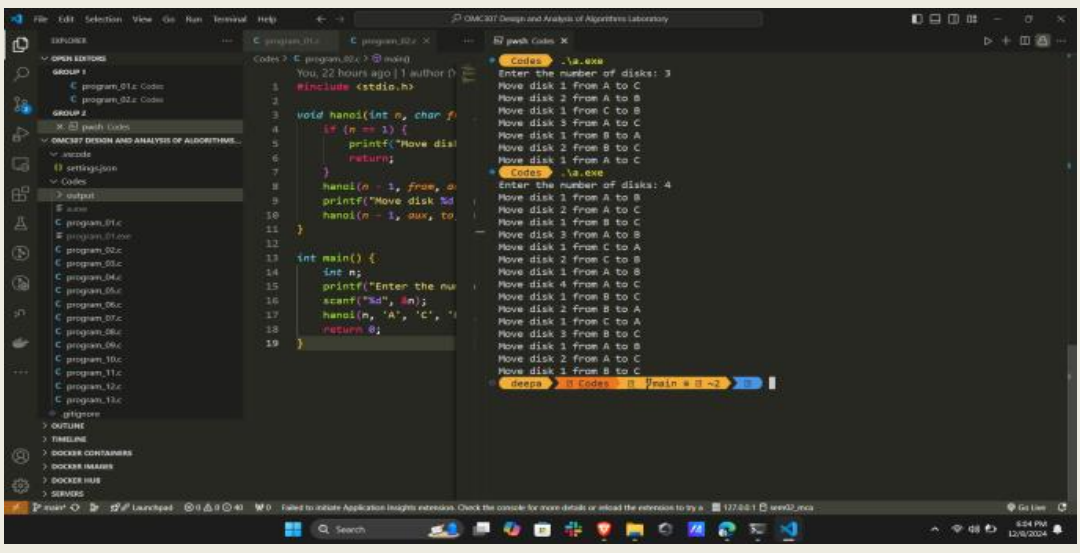
2. Write a C program using recursive function to implement towers of Hanoi problem.

Program 02 : Towers of Hanoi

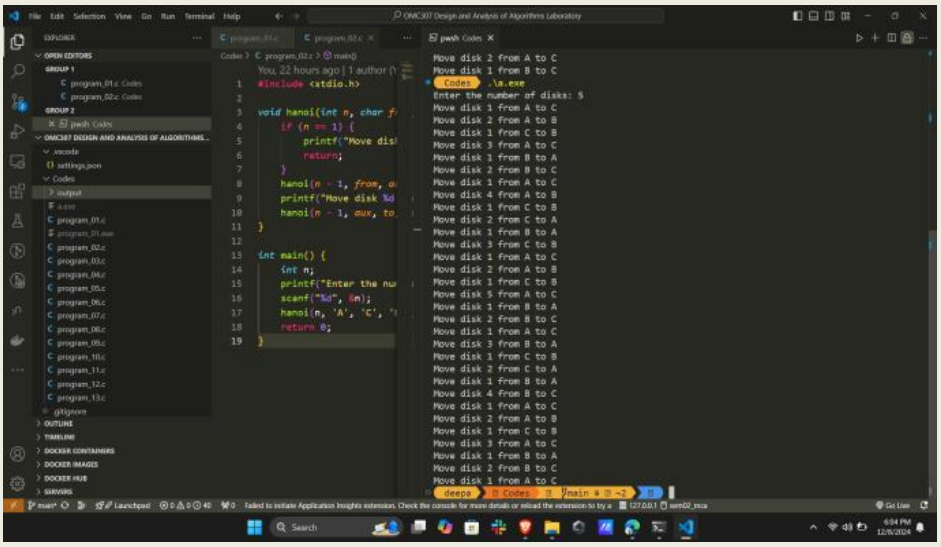
```c
#include <stdio.h>
void hanoi (int n, char from, char to, char x){
    if(n==1){printf("Move disk from %c to %c\n", from, to); return;}
    hanoi(n-1, from, aux, to);
    printf("Move disk %d from %c to %c\n", n, from, to);
    hanoi(n-1, aux, to from);
}
int main(){
    int n;
    printf("Enter the #disks: ");
    scanf("%d", &n);
    hanoi(n, 'A', 'C', 'B');
    return 0;
}
```

Output





Output


#include <stdio.h>
void hanoi(int n, char f
if (n == 1) {
printf("Move dis
return;
}
hanoi(n - 1, from, a
printf("Move disk %d
hanoi(n - 1, aux, to
}
int main() {
int n;
printf("Enter the nu
scanf("%d", &n);
hanoi(n, 'A', 'C', '
return 0;
}
Move disk 2 from A to C
Move disk 1 from B to C
Codes> .\a.exe
Enter the number of disks: 5
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C

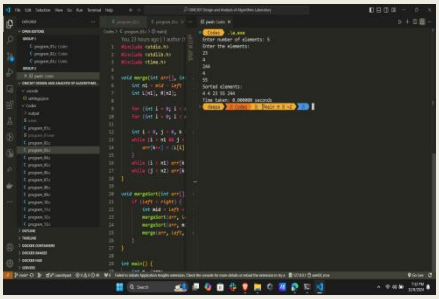
3. Sort a given set of numbers using the merge sort method and determine the time required to sort them.

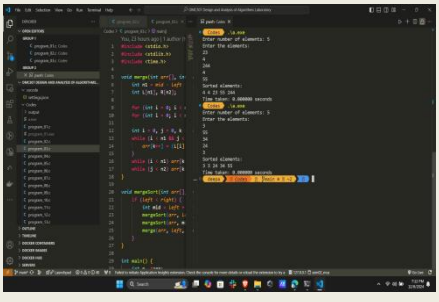
Program03 : Merge sort

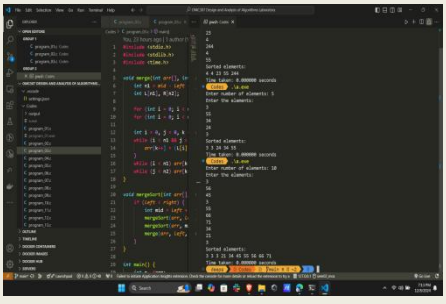
Name : Deepankar Sharma
course : MCA
student ID : 2335512013

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1, n2 = right - mid;
    int L[n1], R[n2];
    for (int i = 0; i < n1; i++) L[i] = arr[left + i];
    for (int i = 0; i < n2; i++) R[i] = arr[mid + 1 + i];
    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) { arr[k++] = (L[i] <= R[j]) ? L[i++] : R[j++]; }
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = L[j++];
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int mid = (left) + (right - left)/2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid+1, right);
        merge(arr, left, mid, right);
    }
}

int main(){
    int n, *arr;
    printf("Enter #elements : ");
    scanf("%d", &n);
    arr = (int *) malloc(n * sizeof(int));
    for(int i=0; i<n; i++) scanf("%d", &arr[i]);
    mergeSort(arr, 0, n-1);
    printf("sorted elements : ");
    for (int i=0; i<n; i++) printf("%d", arr[i]);
    return 0;
}
```

Output







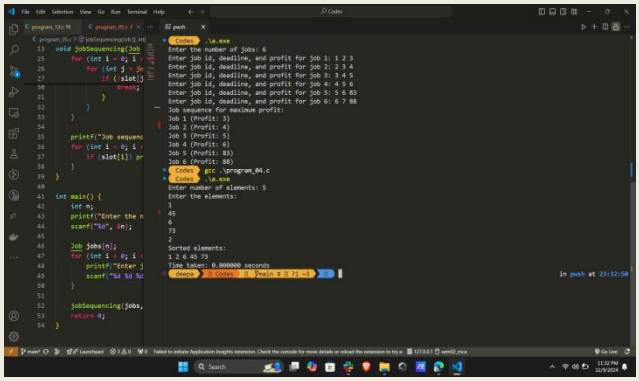
4. Sort a given set of numbers using the quick sort method and determine the time required to sort them.

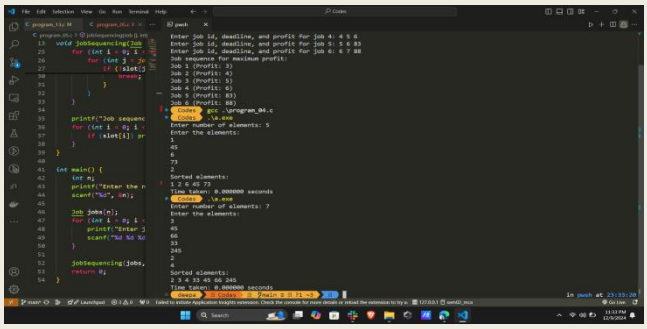
Program 04 : Merge Sort (Quick Sort)

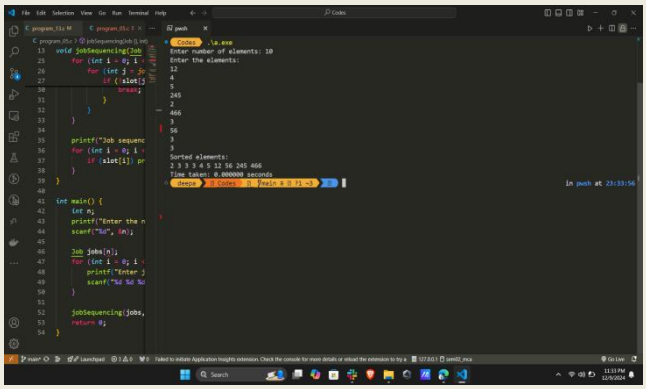
Name : Deepankar Sharma
Course : MCA
Student ID : 23351213

```
#include <stdio.h>
#include <stdlib.h>
int partition (int arr[], int low, int high){
    int pivot = arr[high], i = low-1, temp;
    for(int j= low; j< high; j++) {
        if(arr[j] < pivot){
            arr[i]= arr[j]; arr[j]= temp;
        }
    }
    temp= arr[++i]; arr[i]= arr[high];
    arr[high]= temp;
    return i;
}
void quickSort (int arr[], int low, int high) {
    if (low <high){
        int pi= partition (arr, low, high);
        quickSort (arr, low, pi-1);
        quickSort (arr, pi+1, high);
    }
}
int main () {
    int n, *arr;
    printf("Enter the # elements: ");
    scanf("%d",&n);

    arr = (int *) malloc (n*sizeof(int));
    printf("Enter the elements");
    for(int i=0; i<n; i++) scanf("%d",&arr[i]);

    quickSort(arr, 0, n-1);

    printf(" sorted elements");
    for (int i=0; i<n; i++) printf("%d", arr[i]);

    return 0;

}
```

Output


Enter the number of jobs: 6
Enter job id, deadline, and profit for job 1: 1 2 3
Enter job id, deadline, and profit for job 2: 2 3 4
Enter job id, deadline, and profit for job 3: 3 4 5
Enter job id, deadline, and profit for job 4: 4 5 6
Enter job id, deadline, and profit for job 5: 5 6 83
Enter job id, deadline, and profit for job 6: 6 7 88
Job sequence for maximum profit:
Job 1 (Profit: 3)
Job 2 (Profit: 4)
Job 3 (Profit: 5)
Job 4 (Profit: 6)
Job 5 (Profit: 83)
Job 6 (Profit: 88)
gcc .\program_04.c
Enter number of elements: 5
Enter the elements:
1
45
6
73
2
Sorted elements:
1 2 6 45 73
Time taken: 0.000000 seconds


Enter job id, deadline, and profit for job 4: 4 5 6
Enter job id, deadline, and profit for job 5: 5 6 83
Enter job id, deadline, and profit for job 6: 6 7 88
Job sequence for maximum profit:
Job 1 (Profit: 3)
Job 2 (Profit: 4)
Job 3 (Profit: 5)
Job 4 (Profit: 6)
Job 5 (Profit: 83)
Job 6 (Profit: 88)
gcc .\program_04.c
Enter number of elements: 5
Enter the elements:
1
45
6
73
2
Sorted elements:
1 2 6 45 73
Time taken: 0.000000 seconds
Enter number of elements: 7
Enter the elements:
1
45
66
33
245
2
4
Sorted elements:
1 2 4 33 45 66 245
Time taken: 0.000000 seconds


Enter number of elements: 10
Enter the elements:
12
4
5
245
2
466
3
56
3
3
Sorted elements:
2 3 3 3 4 5 12 56 245 466
Time taken: 0.000000 seconds

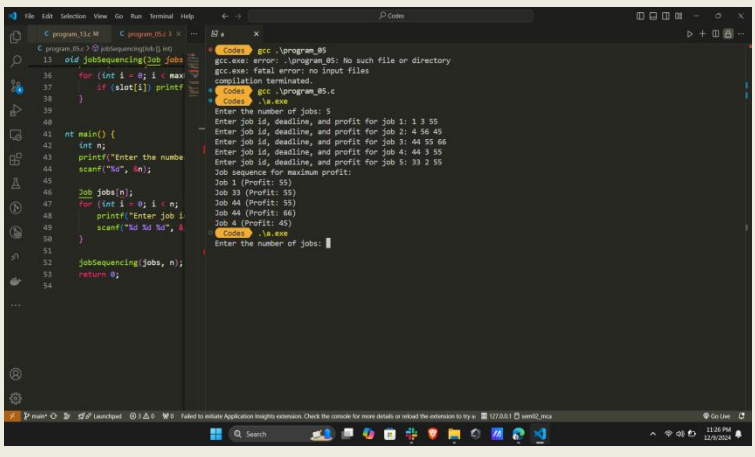
5. Given a set of n jobs, with (d_i) and (p_i) as deadline and profit for job i, write a program to find an optimal sequence of jobs, i.e., find the sequence of jobs that can be completed by their deadline and giving a maximum profit. Estimate the time complexity of the algorithm.

Program 05: Job Sequencing Problem

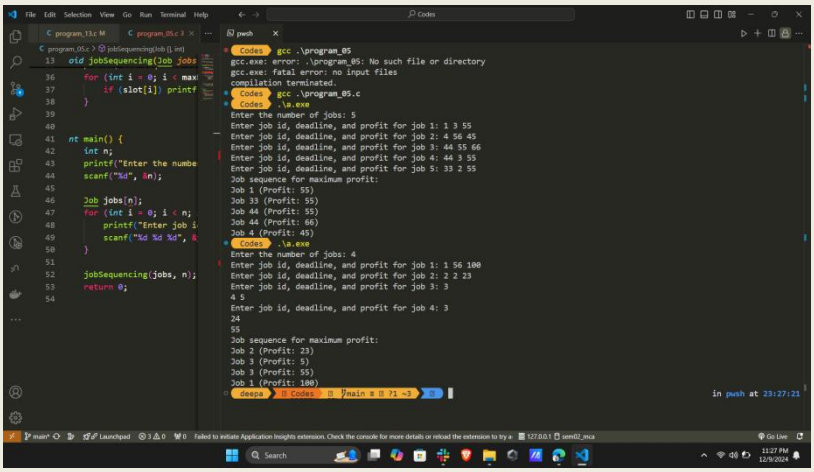
Name: Deepankar Sharma
Course: MCA
Student ID: 233512013

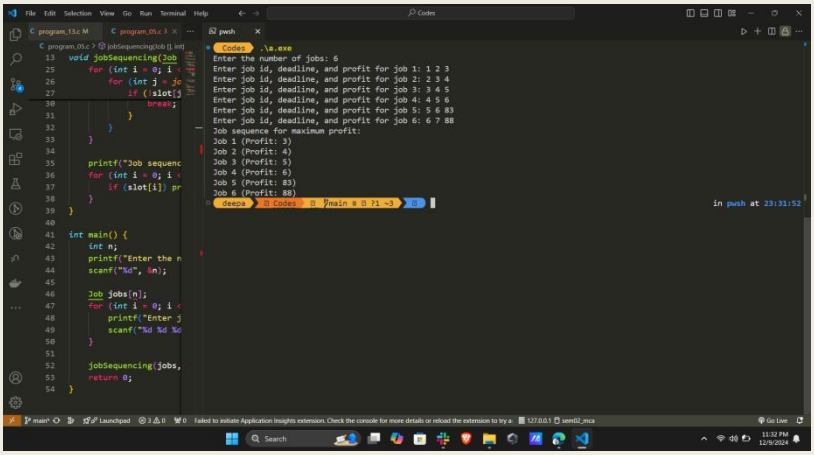
```
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
typedef struct {int id,deadline,profit;}Job;
int compare(const void *a, const void *b){
    return ((Job*)b)->profit - ((Job*)a)->profit;
}

void jobSequencing(Job jobs[], int n){
    qsort(jobs,n,sizeof(Job),compare);
    int maxDeadline=0;
    for(int i=0;i<n;i++){
        if(jobs[i].deadline > maxDeadline) maxDeadline=jobs[i].
                                                    deadline;
    }
    bool slot[maxDeadline];
    int result[maxDeadline];
    for(int i=0;i<maxDeadline;i++) slot[i]=false;
    for(int i=0; i<n; i++){
        for(int j=jobs[i].deadline-1;j>=0;j--){
            if(!slot[j]){
                slot[j]=true; result[j]=i; break;
            }
        }
    }
    printf("Job sequence for maximum profit: ");
    for(int i=0;i<maxDeadline;i++){
        if(slot[i]) printf("Job %d (Profit %d)\n",jobs[results[i]].id,
                                jobs[results[i]].profit);
    }
}
```

Output



Output


gcc .\program_05
gcc.exe: error: .\program_05: No such file or directory
gcc.exe: fatal error: no input files
compilation terminated.
gcc .\program_05.c
.\a.exe
Enter the number of jobs: 5
Enter job id, deadline, and profit for job 1: 1 3 55
Enter job id, deadline, and profit for job 2: 4 56 45
Enter job id, deadline, and profit for job 3: 44 55 66
Enter job id, deadline, and profit for job 4: 44 3 55
Enter job id, deadline, and profit for job 5: 33 2 55
Job sequence for maximum profit:
Job 1 (Profit: 55)
Job 33 (Profit: 55)
Job 44 (Profit: 55)
Job 44 (Profit: 66)
Job 4 (Profit: 45)
.\a.exe
Enter the number of jobs: 4
Enter job id, deadline, and profit for job 1: 1 56 100
Enter job id, deadline, and profit for job 2: 2 2 23
Enter job id, deadline, and profit for job 3: 3
4 5
Enter job id, deadline, and profit for job 4: 3
24
55
Job sequence for maximum profit:
Job 2 (Profit: 23)
Job 3 (Profit: 5)
Job 3 (Profit: 55)
Job 1 (Profit: 100)
deepa
in pwsh at 23:27:21


.\a.exe
Enter the number of jobs: 6
Enter job id, deadline, and profit for job 1: 1 2 3
Enter job id, deadline, and profit for job 2: 2 3 4
Enter job id, deadline, and profit for job 3: 3 4 5
Enter job id, deadline, and profit for job 4: 4 5 6
Enter job id, deadline, and profit for job 5: 5 6 83
Enter job id, deadline, and profit for job 6: 6 7 88
Job sequence for maximum profit:
Job 1 (Profit: 3)
Job 2 (Profit: 4)
Job 3 (Profit: 5)
Job 4 (Profit: 6)
Job 5 (Profit: 83)
Job 6 (Profit: 88)
deepa
in pwsh at 23:31:52

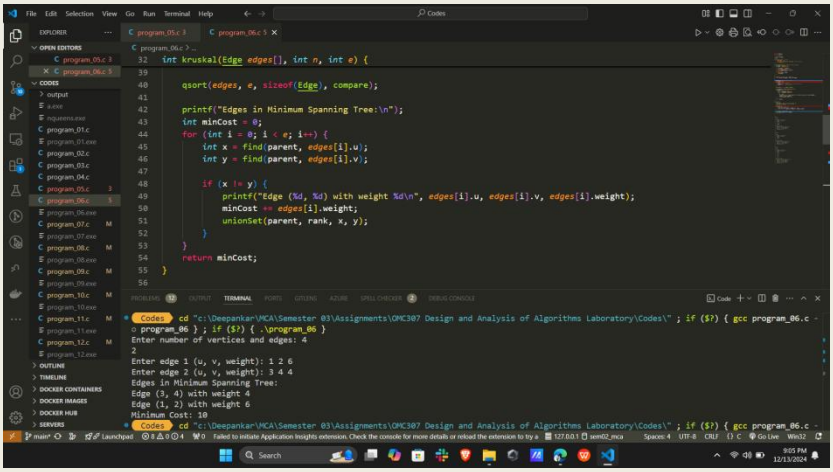
6. Write C programs to find the minimum cost spanning tree using Kruskal's method.

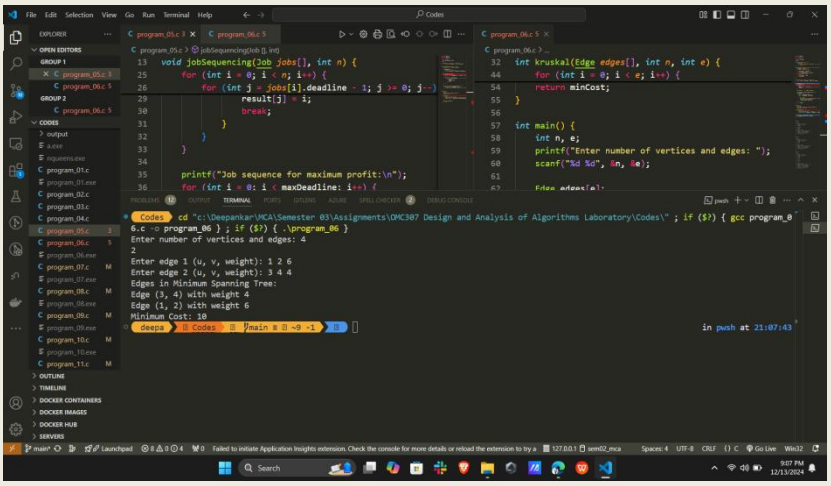
Program 06 : Minimum Spanning Tree

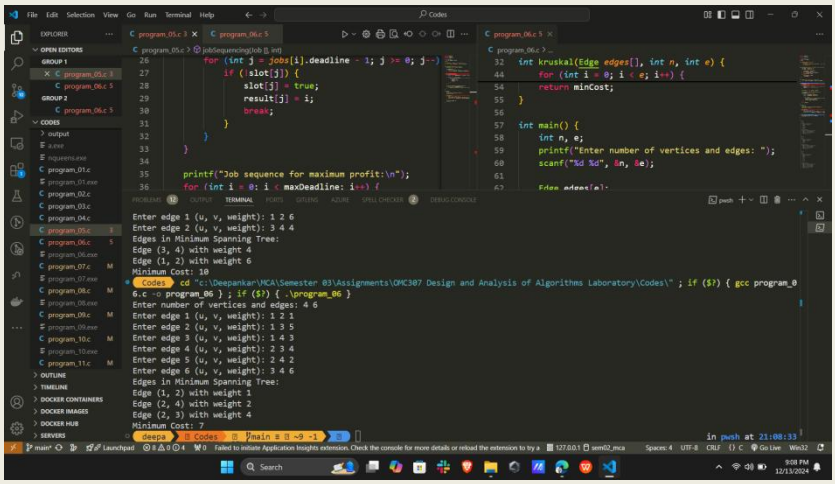
Name : Deepankar Sharma
Course : MCA
Student ID : 23351201 3

```
#include <stdio.h>
#include <stdlib.h>
typedef struct { int u, v, weight; } Edge;
int find (int parent [], int i) {
        if (parent [i] == -1) return i;
        return find(parent , parent [i]);
}
void unionSet ( int parent [], int x, int y) { parent[x] = y; }
int compare (const void * a, const void *b) {
    return ((Edge*) a) -> weight -((Edge *) b) -> weight ;
 }
void Kruskal(Edge edges[] , int n , int e) {
    int parent[n];
    for (int i =0; i <n ; i ++) parent[i] = -1;
    qsort (edges , e , sizeof (Edge), compare);
    printf ("Edges in minimum spanning tree : ");
    int minCost=0;
    for (int i =0; i<e ; i++) {
        int x = find (parent, edges[i] . u);
        int y = find(parent , edges [i] . v);
        if(x!=y) {
            printf ("Edge(%d, %d) with (%d) weight \n", edges[i]. u,
                    edges [i] . v , edges [i] . weight);
            minCost += edges[i] . weight ; unionSet (parent , x, y) ;
        }
    }
    printf ("Minimum cost : %d \n" , minCost);
}
int main () {
    int n,e; printf ("Enter # vertices and # edges :"); scanf ("%d %d",&n,&e);
    Edge edges [e];
    for (int i =0; i<e ; i++) {
        printf ("Enter edge %d (u, v, weight) : " , i +1);
        scanf ("%d %d %d ", & edges [i]. u , & edges [i]. v , &edges[i].
                                                    weight);
    kruskal (edges, n, e);
    return 0;
}
```

Output


int kruskal(Edge edges[], int n, int e) {
qsort(edges, e, sizeof(Edge), compare);
printf("Edges in Minimum Spanning Tree:\n");
int minCost = 0;
for (int i = 0; i < e; i++) {
int x = find(parent, edges[i].u);
int y = find(parent, edges[i].v);
if (x != y) {
printf("Edge (%d, %d) with weight %d\n", edges[i].u, edges[i].v, edges[i].weight);
minCost += edges[i].weight;
unionSet(parent, rank, x, y);
}
}
return minCost;
}
Enter number of vertices and edges: 4
2
Enter edge 1 (u, v, weight): 1 2 6
Enter edge 2 (u, v, weight): 3 4 4
Edges in Minimum Spanning Tree:
Edge (3, 4) with weight 4
Edge (1, 2) with weight 6
Minimum Cost: 10


void jobSequencing(Job jobs[], int n) {
printf("Job sequence for maximum profit:\n");
int kruskal(Edge edges[], int n, int e) {
int main() {
int n, e;
printf("Enter number of vertices and edges: ");
scanf("%d %d", &n, &e);
Enter number of vertices and edges: 4
2
Enter edge 1 (u, v, weight): 1 2 6
Enter edge 2 (u, v, weight): 3 4 4
Edges in Minimum Spanning Tree:
Edge (3, 4) with weight 4
Edge (1, 2) with weight 6
Minimum Cost: 10


if (!slot[j]) {
slot[j] = true;
result[j] = i;
break;
printf("Job sequence for maximum profit:\n");
int kruskal(Edge edges[], int n, int e) {
int main() {
int n, e;
printf("Enter number of vertices and edges: ");
scanf("%d %d", &n, &e);
Enter edge 1 (u, v, weight): 1 2 6
Enter edge 2 (u, v, weight): 3 4 4
Edges in Minimum Spanning Tree:
Edge (3, 4) with weight 4
Edge (1, 2) with weight 6
Minimum Cost: 10
Enter number of vertices and edges: 4 6
Enter edge 1 (u, v, weight): 1 2 1
Enter edge 2 (u, v, weight): 1 3 5
Enter edge 3 (u, v, weight): 1 4 3
Enter edge 4 (u, v, weight): 2 3 4
Enter edge 5 (u, v, weight): 2 4 2
Enter edge 6 (u, v, weight): 3 4 6
Edges in Minimum Spanning Tree:
Edge (1, 2) with weight 1
Edge (2, 4) with weight 2
Edge (2, 3) with weight 4
Minimum Cost: 7

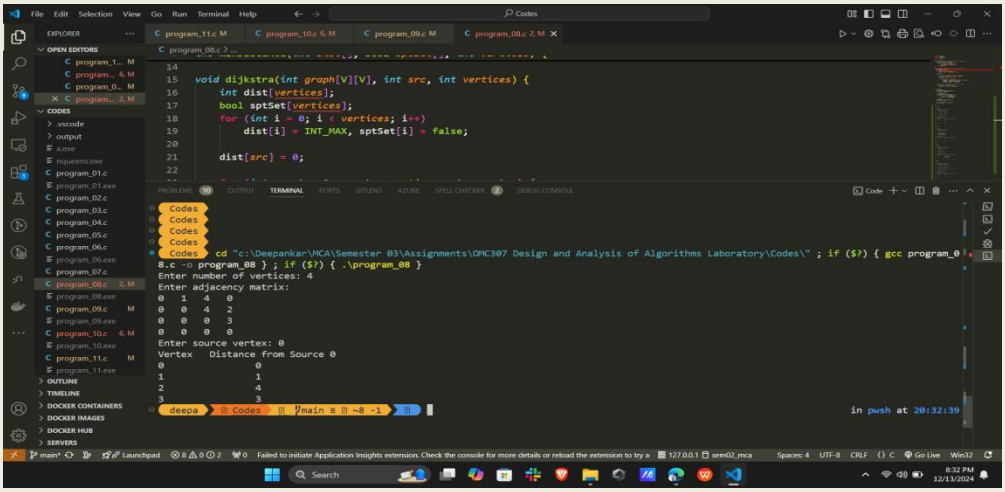
7. Write a C program to implement Dijkstra's Algorithm to find the shortest paths from a given source vertex to all other vertices.

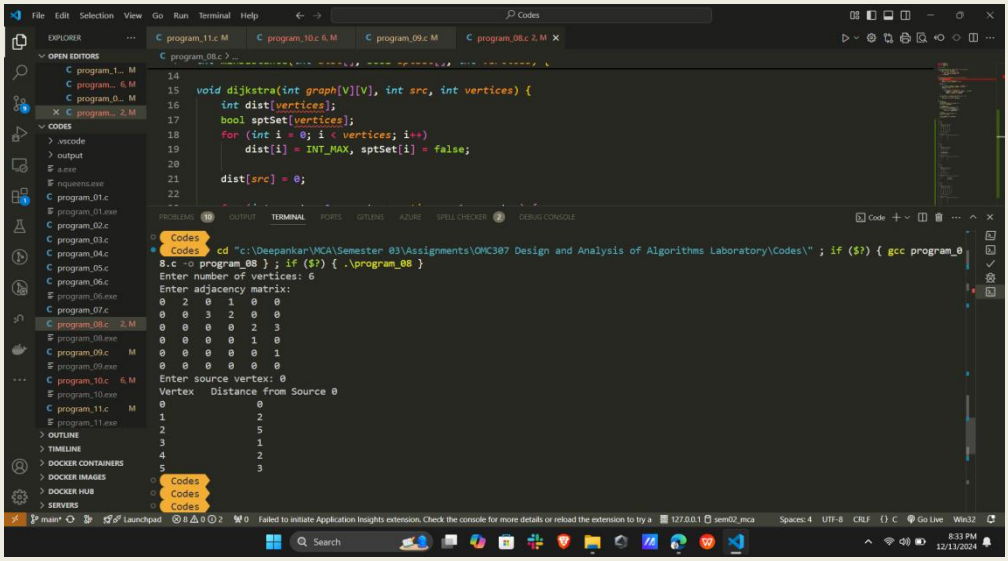
Program 08 : Dijkstra's Algorithm

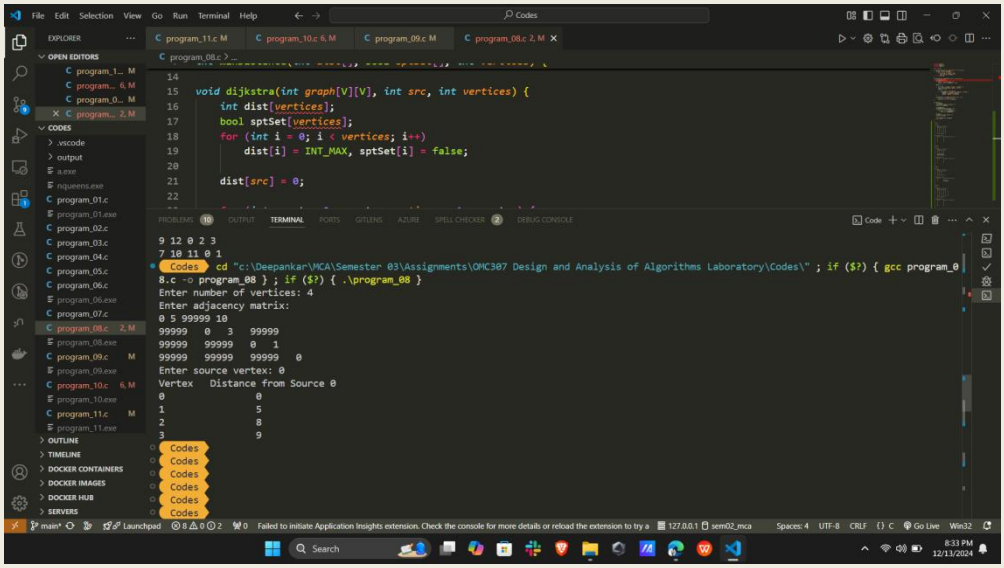
Name : Deepankar Sharma
Course : MCA
Student ID : 23351201 3

```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#define V 100
int minDistance(int dist[], bool sptSet[], int vertices) {
    int min = INT_MAX, min_index;
    for (int v=0; v<vertices; v++) if (!sptSet[v] && dist[v]<=min)
            min = dist[v], min_index=v;
    return min_index;
}
void dijkstra(int graph[V][V], int src, int vertices) {
    int dist[vertices]; bool sptSet[vertices];
    for(int i=0; i<vertices; i++) dist[i]= INT_MAX, sptSet[i]= false;
    dist[src] = 0;
    for(int count=0; count<vertices-1; count++) {
        int u= minDistance(dist, sptSet, vertices);
        sptSet[u]= true;
    }
    for(int v=0; v<vertices; v++)
        if( !sptSet[v] && graph[u][v] && dist[u]!=INT_MAX &&
            dist[u]+graph[u][v]<dist[v]) dist[v]=dist[u]+graph[u][v];
    }
    printf("Vertex \t Distance from Source %d \n", src);
    for(int i=0; i<vertices; i++) printf("%d \t\t %d\n", i, dist[i]);
}
int main() {
    int vertices; int src;
    printf("Enter # vertices: "); scanf("%d", &vertices);
    int graph[V][V];
    printf("Enter adjacency matrix: \n");
    for(int i=0; i<vertices; i++)
        for(int j=0; j<vertices; j++)
            scanf("%d", &graph[i][j]);
    scanf("%d", &src);
    dijstra(graph, src, vertices);
    return 0;
}
```

Output


void dijkstra(int graph[V][V], int src, int vertices) {
int dist[vertices];
bool sptSet[vertices];
for (int i = 0; i < vertices; i++)
dist[i] = INT_MAX, sptSet[i] = false;
dist[src] = 0;
cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_08.c -o program_08 } ; if ($?) { .\program_08 }
Enter number of vertices: 4
Enter adjacency matrix:
0 1 4 0
0 0 4 2
0 0 0 3
0 0 0 0
Enter source vertex: 0
Vertex Distance from Source 0
0 0
1 1
2 4
3 3


void dijkstra(int graph[V][V], int src, int vertices) {
int dist[vertices];
bool sptSet[vertices];
for (int i = 0; i < vertices; i++)
dist[i] = INT_MAX, sptSet[i] = false;
dist[src] = 0;
cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_08.c -o program_08 } ; if ($?) { .\program_08 }
Enter number of vertices: 6
Enter adjacency matrix:
0 2 0 1 0 0
0 0 3 2 0 0
0 0 0 0 2 3
0 0 0 0 1 0
0 0 0 0 0 1
0 0 0 0 0 0
Enter source vertex: 0
Vertex Distance from Source 0
0 0
1 2
2 5
3 1
4 2
5 3


void dijkstra(int graph[V][V], int src, int vertices) {
int dist[vertices];
bool sptSet[vertices];
for (int i = 0; i < vertices; i++)
dist[i] = INT_MAX, sptSet[i] = false;
dist[src] = 0;
9 12 0 2 3
7 10 11 0 1
cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_08.c -o program_08 } ; if ($?) { .\program_08 }
Enter number of vertices: 4
Enter adjacency matrix:
0 5 99999 10
99999 0 3 99999
99999 99999 0 1
99999 99999 99999 0
Enter source vertex: 0
Vertex Distance from Source 0
0 0
1 5
2 8
3 9

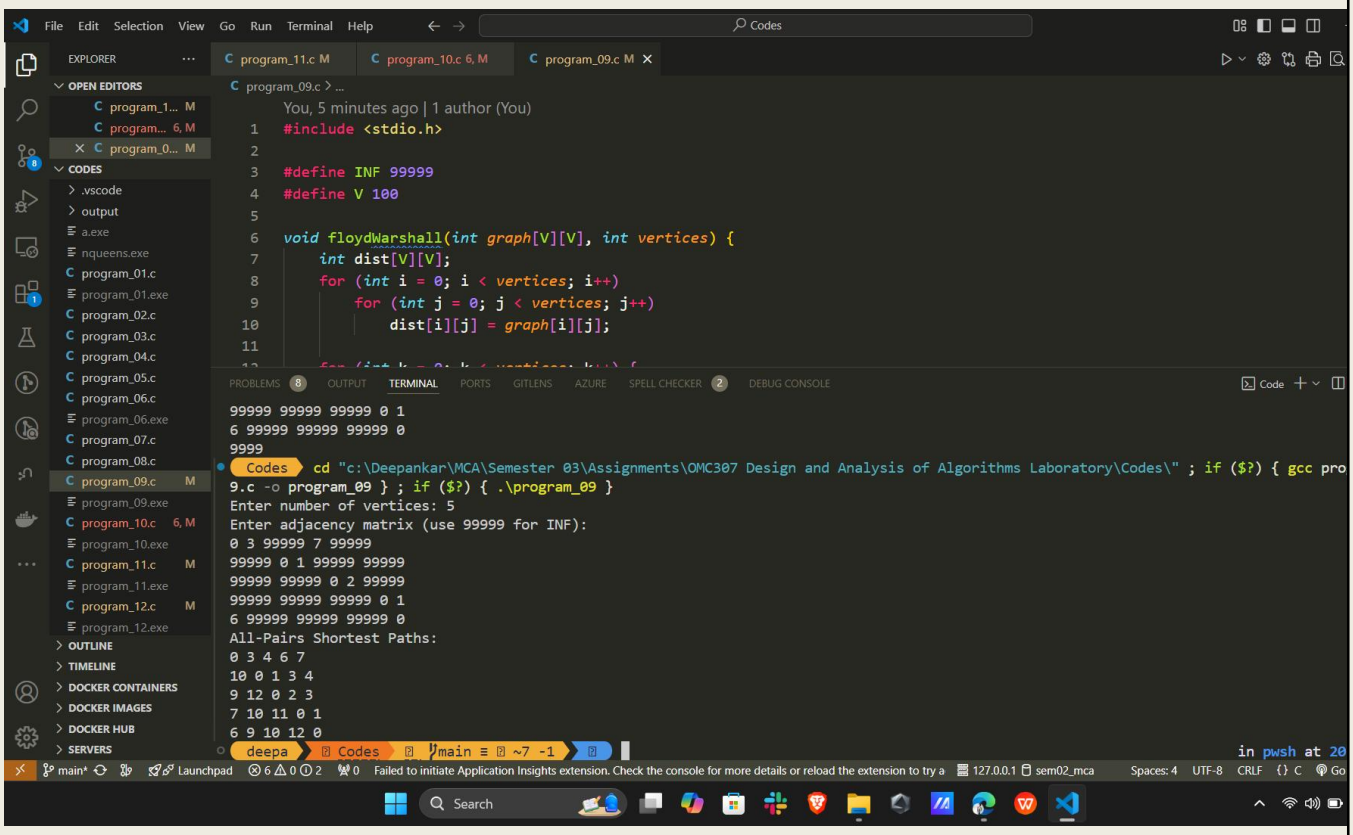
8. Write C program to find all pairs of shortest paths using Floyd's algorithm.

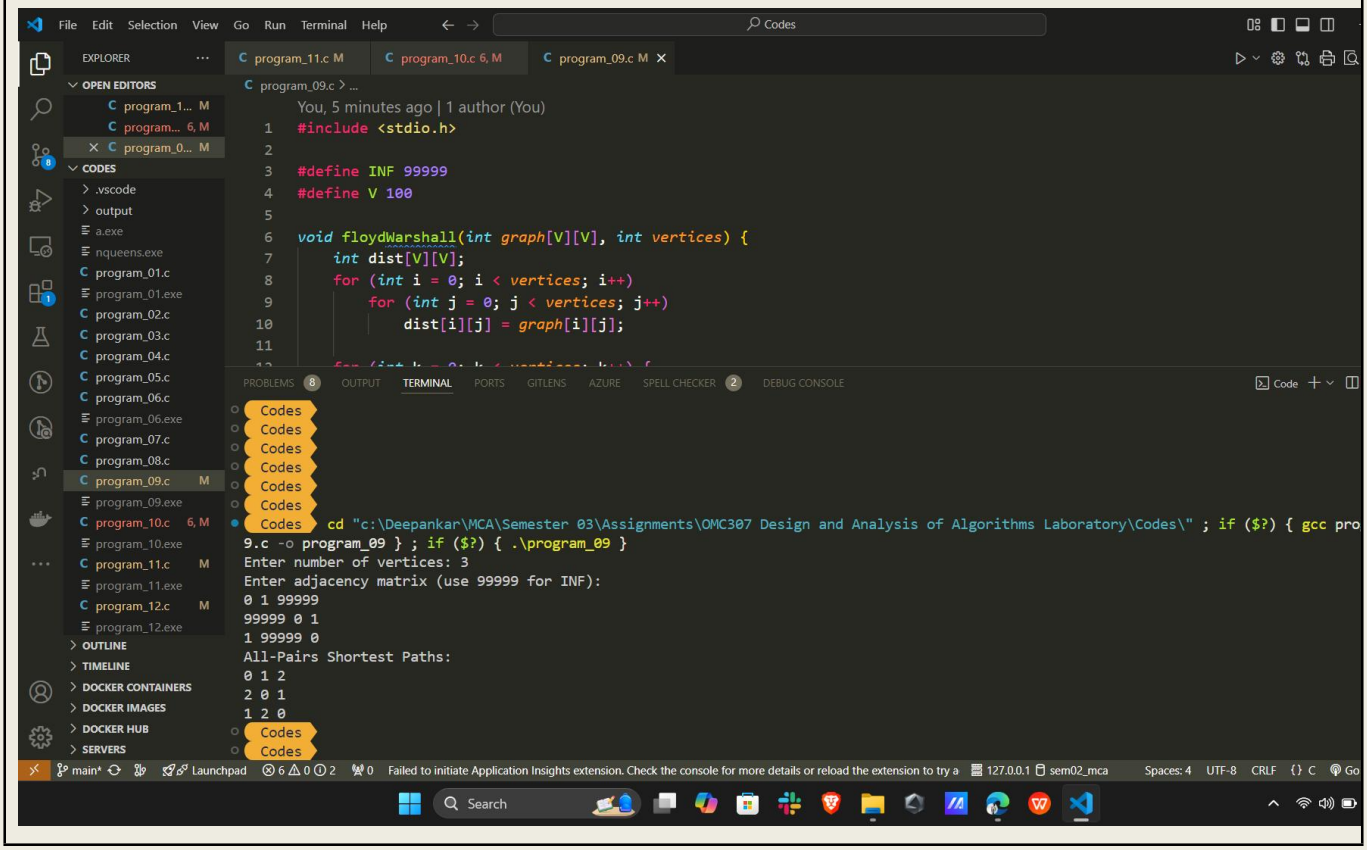
Program 09 : Floyd's Algorithm

Name : Deepankar Sharma
course : MCA
student ID : 2335512015

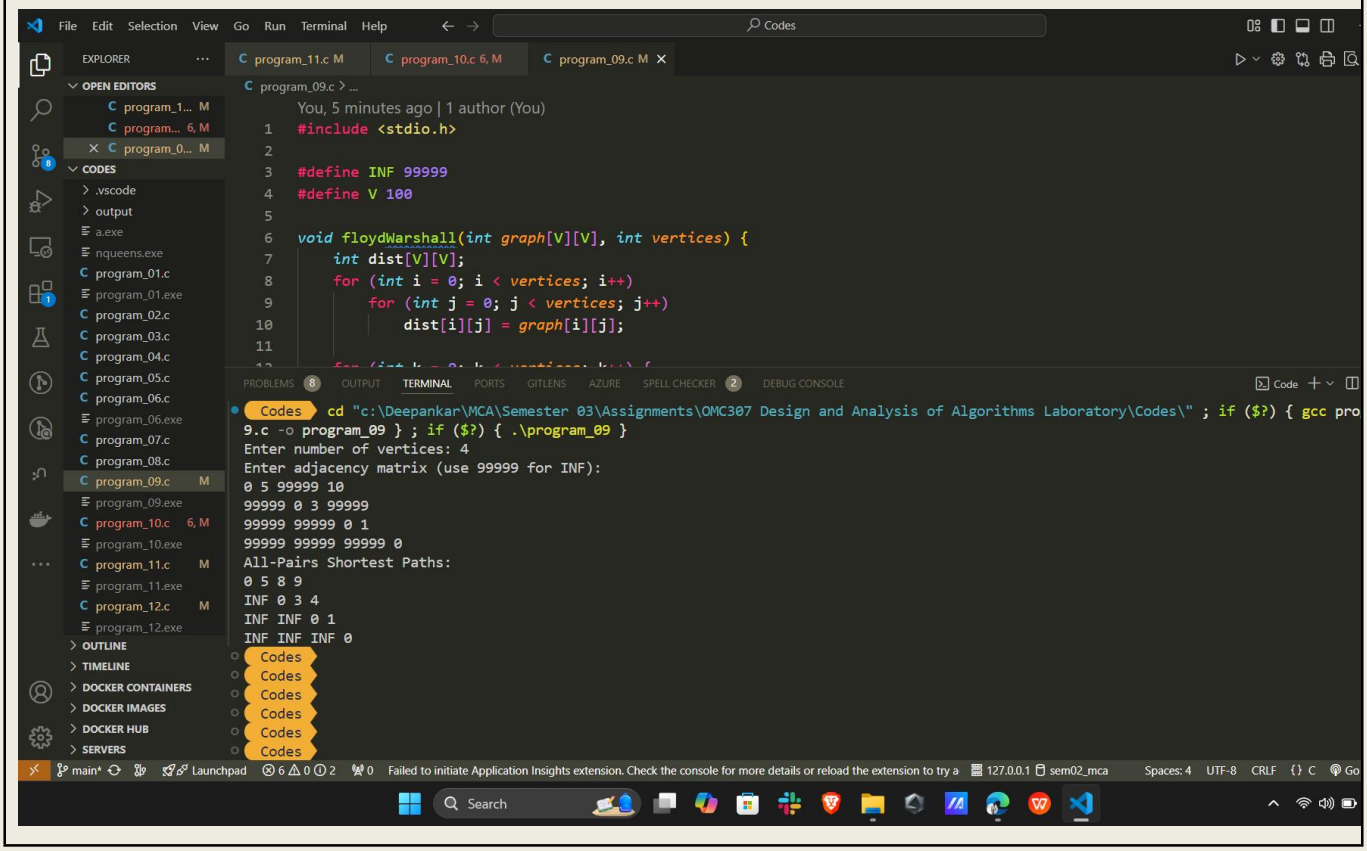
```
#include <stdio.h>
#define INF 99999
#define V 100

void floydWarshall (int graph[V][V], int vertices) {
    int dist[V][V];
    for (int i=0; i<vertices; i++) for (int j=0; j<vertices; j++)
      dist[i][j]= graph[i][j];
    for (int k=0; k<vertices; k++) for (int i=0; i<vertices; i++)
     for(int j=0; j<vertices; j++) {
           if(dist[i][k] + dist[k][j] < dist[i][j])
              dist[i][j]= dist[i][k] + dist[k][j];

    printf("All pairs shortest paths :\n");
    for(int i=0; i<vertices; i++){
        for(int j=0; j<vertices; j++) {
            if (dist[i][j] == INF)
                printf("INF ");
            else
                printf("%d ", dist[i][j]);
          } printf("\n");

    }
}
int main() {
    int vertices;
    printf("Enter # vertices "); scanf("%d",&vertices);
    int graph[V][V];
    printf("Enter adjacency matrix");
    for (int i=0; i<vertices; i++)
        for(int j=0; j<vertices; j++)
            scanf("%d", &graph[i][j]);
    floydWarshall(graph,vertices);
    return 0;
}
```

Output


File Edit Selection View Go Run Terminal Help
Codes
program_11.c M
program_10.c 6, M
program_09.c M
You, 5 minutes ago | 1 author (You)
#include <stdio.h>
#define INF 99999
#define V 100
void floydWarshall(int graph[V][V], int vertices) {
int dist[V][V];
for (int i = 0; i < vertices; i++)
for (int j = 0; j < vertices; j++)
dist[i][j] = graph[i][j];
PROBLEMS 8 OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER 2 DEBUG CONSOLE
99999 99999 99999 0 1
6 99999 99999 99999 0
9999
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc pro
9.c -o program_09 } ; if ($?) { .\program_09 }
Enter number of vertices: 5
Enter adjacency matrix (use 99999 for INF):
0 3 99999 7 99999
99999 0 1 99999 99999
99999 99999 0 2 99999
99999 99999 99999 0 1
6 99999 99999 99999 0
All-Pairs Shortest Paths:
0 3 4 6 7
10 0 1 3 4
9 12 0 2 3
7 10 11 0 1
6 9 10 12 0
deepa Codes main ~7 -1
Launchpad
Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a
Search


File Edit Selection View Go Run Terminal Help
Codes
program_11.c M
program_10.c 6, M
program_09.c M
You, 5 minutes ago | 1 author (You)
#include <stdio.h>
#define INF 99999
#define V 100
void floydWarshall(int graph[V][V], int vertices) {
int dist[V][V];
for (int i = 0; i < vertices; i++)
for (int j = 0; j < vertices; j++)
dist[i][j] = graph[i][j];
PROBLEMS 8 OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER 2 DEBUG CONSOLE
Codes
Codes
Codes
Codes
Codes
Codes
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc pro
9.c -o program_09 } ; if ($?) { .\program_09 }
Enter number of vertices: 3
Enter adjacency matrix (use 99999 for INF):
0 1 99999
99999 0 1
1 99999 0
All-Pairs Shortest Paths:
0 1 2
2 0 1
1 2 0
Codes
Codes
Launchpad
Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a
Search

Output


File Edit Selection View Go Run Terminal Help
Codes
EXPLORER
program_11.c M
program_10.c 6, M
program_09.c M
program_09.c > ...
You, 5 minutes ago | 1 author (You)
#include <stdio.h>
#define INF 99999
#define V 100
void floydWarshall(int graph[V][V], int vertices) {
int dist[V][V];
for (int i = 0; i < vertices; i++)
for (int j = 0; j < vertices; j++)
dist[i][j] = graph[i][j];
OPEN EDITORS
CODES
.vscode
output
a.exe
nqueens.exe
program_01.c
program_01.exe
program_02.c
program_03.c
program_04.c
program_05.c
program_06.c
program_06.exe
program_07.c
program_08.c
program_09.c
program_09.exe
program_10.c
program_10.exe
program_11.c
program_11.exe
program_12.c
program_12.exe
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
SERVERS
PROBLEMS 8
OUTPUT
TERMINAL
PORTS
GITLENS
AZURE
SPELL CHECKER 2
DEBUG CONSOLE
Code
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc pro
9.c -o program_09 } ; if ($?) { .\program_09 }
Enter number of vertices: 4
Enter adjacency matrix (use 99999 for INF):
0 5 99999 10
99999 0 3 99999
99999 99999 0 1
99999 99999 99999 0
All-Pairs Shortest Paths:
0 5 8 9
INF 0 3 4
INF INF 0 1
INF INF INF 0
Codes
Codes
Codes
Codes
Codes
Codes
main*
Launchpad
Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a
127.0.0.1
sem02_mca
Spaces: 4
UTF-8
CRLF
C
Go
Search

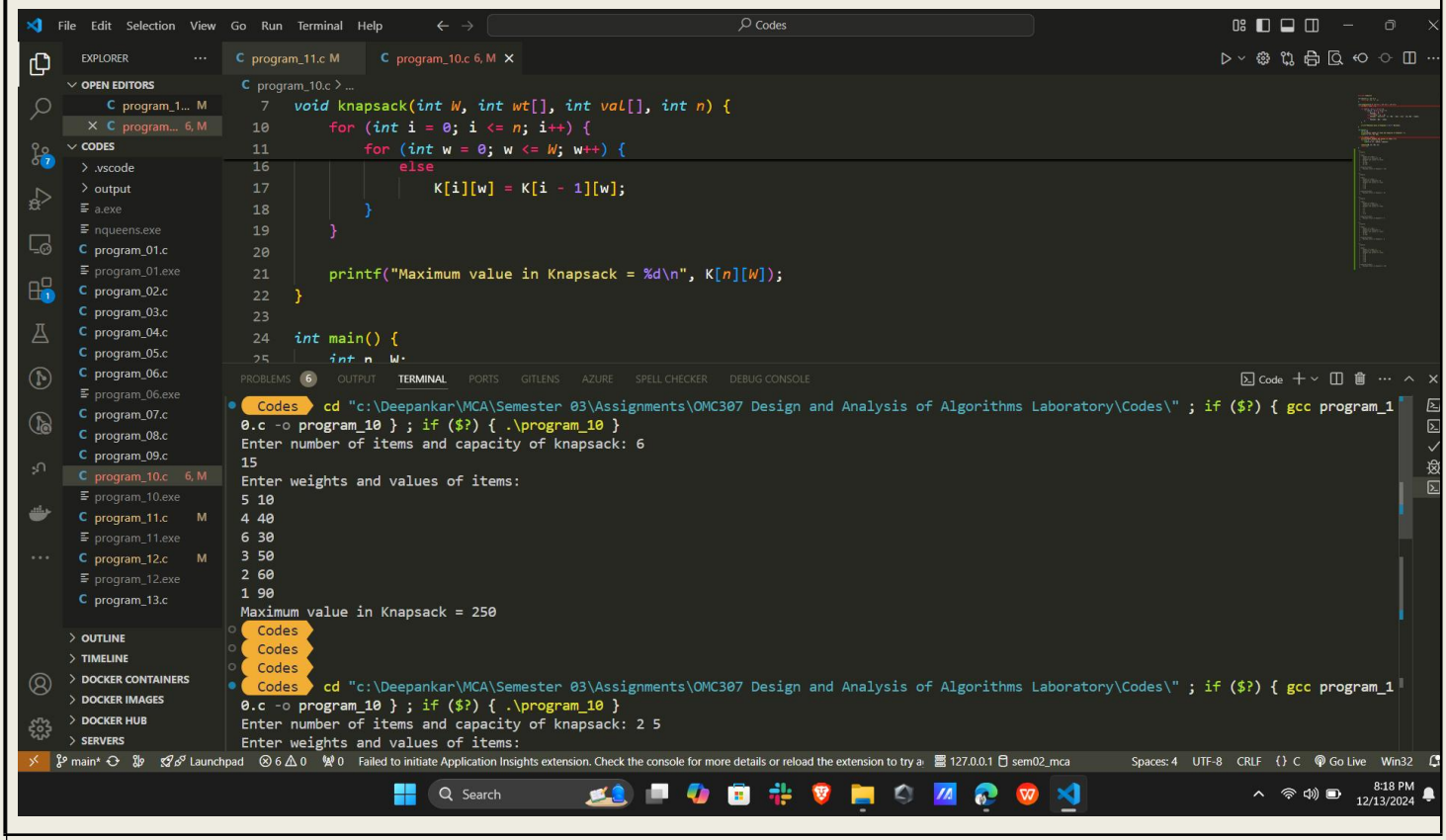
9. Write C program to implement 0/1 knapsack problem using dynamic programming.

Program 10 : 0-1 knapsack problem

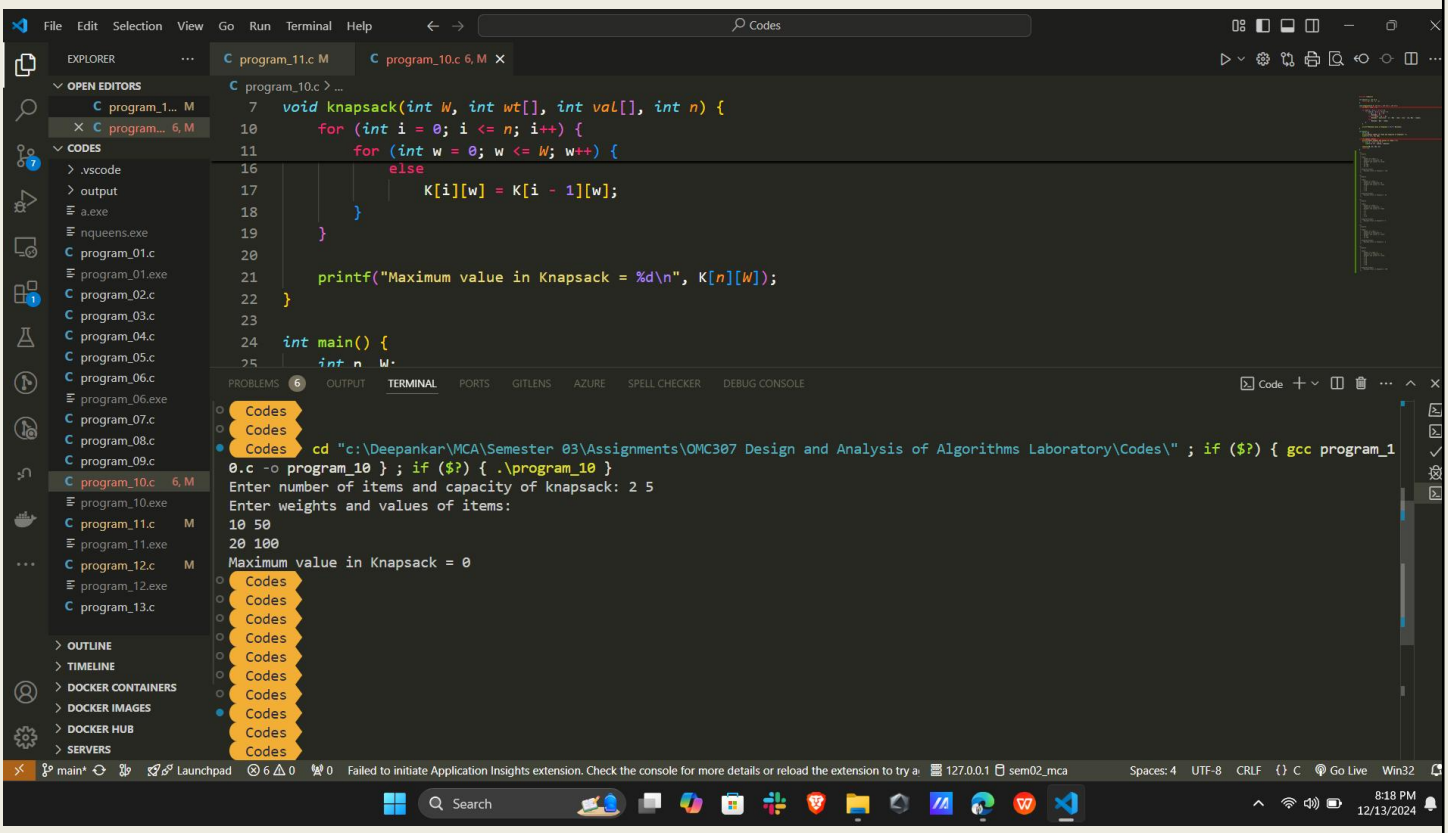
Name : Deepankar Sharma
Course : MCA

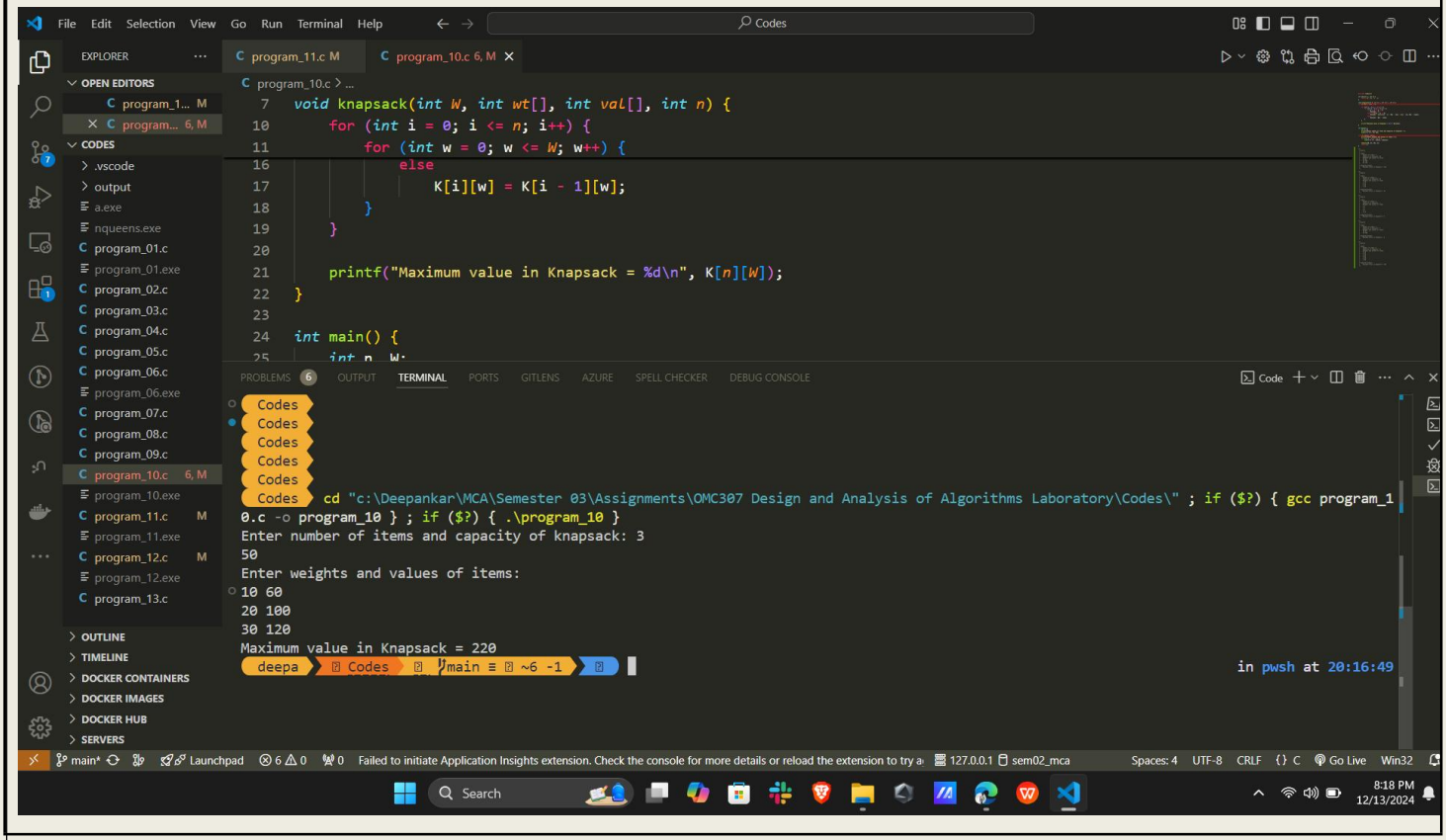
```
#include <stdio.h>
int max(int a, int b) { return (a > b)? a : b;}
void knapsack(int W, int wt[], int val[], int n){
    int K[n+1][W+1];
    for (int i=0; i<=n; i++) for(int w=0; w<=W ; w++){
        if(i==0 || w==0) K[i][w]=0;
        else if(wt[i-1] <= w) K[i][w]=max(K[i-1][w],
                                  val[i-1] + K[i-1][w-wt[i-1]]);
        else K[i][w] = K[i-1][w];
    }
    printf("maximum value in knapsack = %d \n", K[n][w]);
}
```

Output

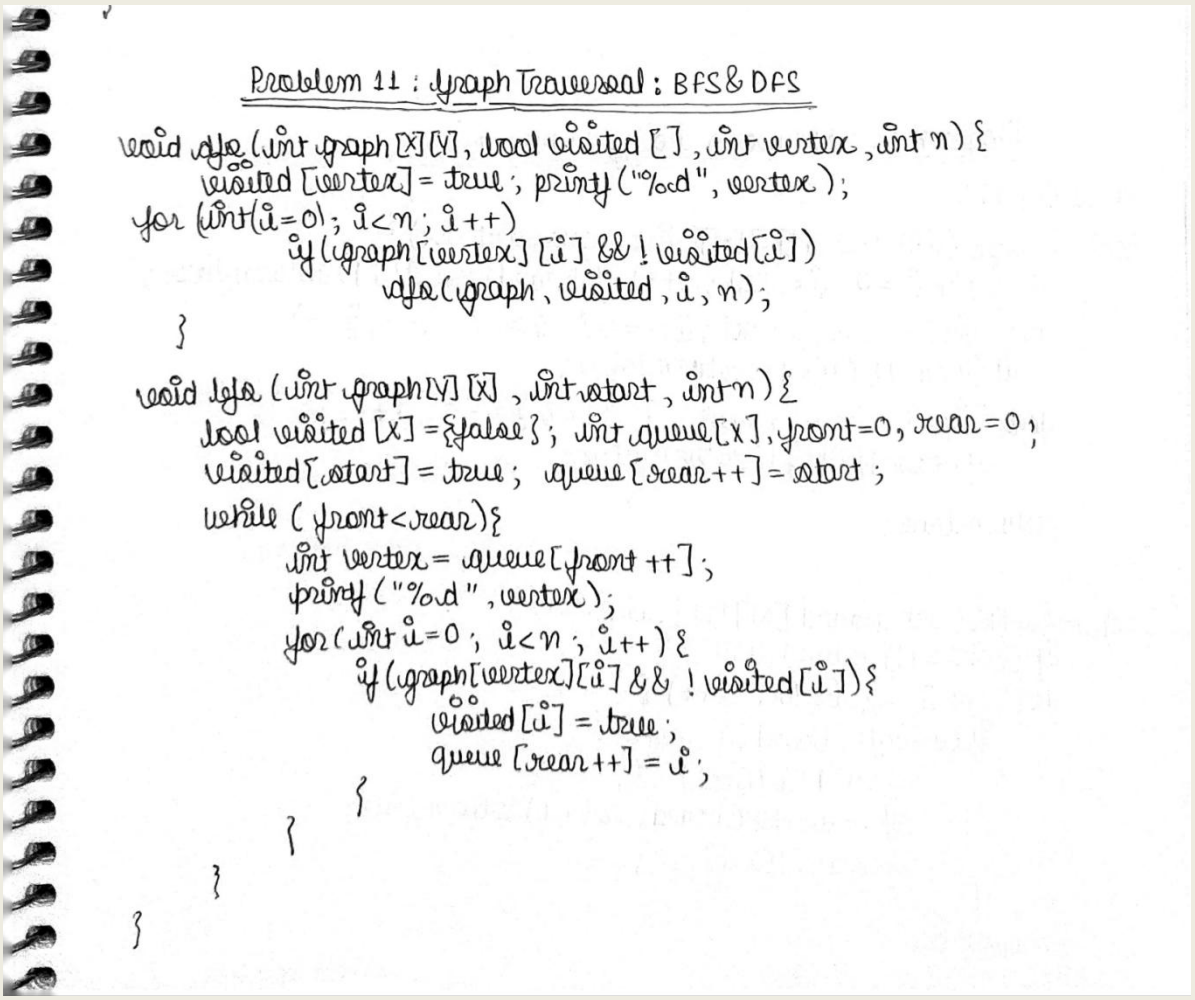


Output


File Edit Selection View Go Run Terminal Help
Codes
EXPLORER
program_11.c M
program_10.c 6, M
OPEN EDITORS
CODES
.vscode
output
a.exe
nqueens.exe
program_01.c
program_01.exe
program_02.c
program_03.c
program_04.c
program_05.c
program_06.c
program_06.exe
program_07.c
program_08.c
program_09.c
program_10.c 6, M
program_10.exe
program_11.c M
program_11.exe
program_12.c M
program_12.exe
program_13.c
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
SERVERS
void knapsack(int W, int wt[], int val[], int n) {
for (int i = 0; i <= n; i++) {
for (int w = 0; w <= W; w++) {
else
K[i][w] = K[i - 1][w];
}
}
printf("Maximum value in Knapsack = %d\n", K[n][W]);
}
int main() {
PROBLEMS 6 OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER DEBUG CONSOLE
Code
Codes
Codes
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
0.c -o program_10 } ; if ($?) { .\program_10 }
Enter number of items and capacity of knapsack: 2 5
Enter weights and values of items:
10 50
20 100
Maximum value in Knapsack = 0
Codes
Codes
Codes
Codes
Codes
Codes
Codes
Codes
Codes
main* Launchpad 6 0 0 Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a 127.0.0.1 sem02_mca Spaces: 4 UTF-8 CRLF C Go Live Win32
Search
8:18 PM
12/13/2024


File Edit Selection View Go Run Terminal Help
Codes
EXPLORER
program_11.c M
program_10.c 6, M
OPEN EDITORS
CODES
.vscode
output
a.exe
nqueens.exe
program_01.c
program_01.exe
program_02.c
program_03.c
program_04.c
program_05.c
program_06.c
program_06.exe
program_07.c
program_08.c
program_09.c
program_10.c 6, M
program_10.exe
program_11.c M
program_11.exe
program_12.c M
program_12.exe
program_13.c
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
SERVERS
void knapsack(int W, int wt[], int val[], int n) {
for (int i = 0; i <= n; i++) {
for (int w = 0; w <= W; w++) {
else
K[i][w] = K[i - 1][w];
}
}
printf("Maximum value in Knapsack = %d\n", K[n][W]);
}
int main() {
PROBLEMS 6 OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER DEBUG CONSOLE
Code
Codes
Codes
Codes
Codes
Codes
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
0.c -o program_10 } ; if ($?) { .\program_10 }
Enter number of items and capacity of knapsack: 3
50
Enter weights and values of items:
10 60
20 100
30 120
Maximum value in Knapsack = 220
deepa Codes main ≡ ~6 -1
in pwsh at 20:16:49
main* Launchpad 6 0 0 Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a 127.0.0.1 sem02_mca Spaces: 4 UTF-8 CRLF C Go Live Win32
Search
8:18 PM
12/13/2024

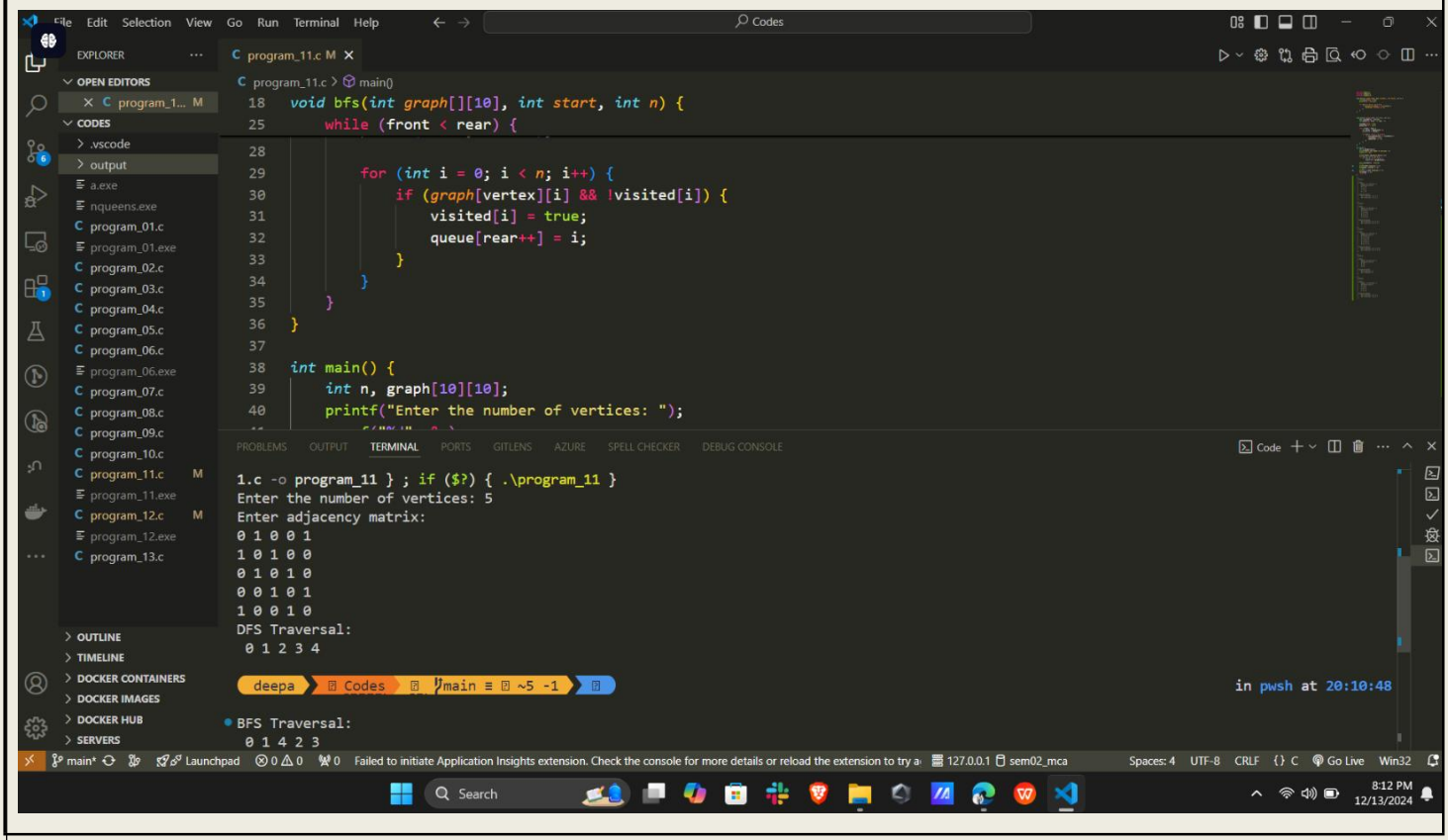
10. Write a C program to perform the following operations on a graph:
    - Depth First Search
    - Breadth First Search



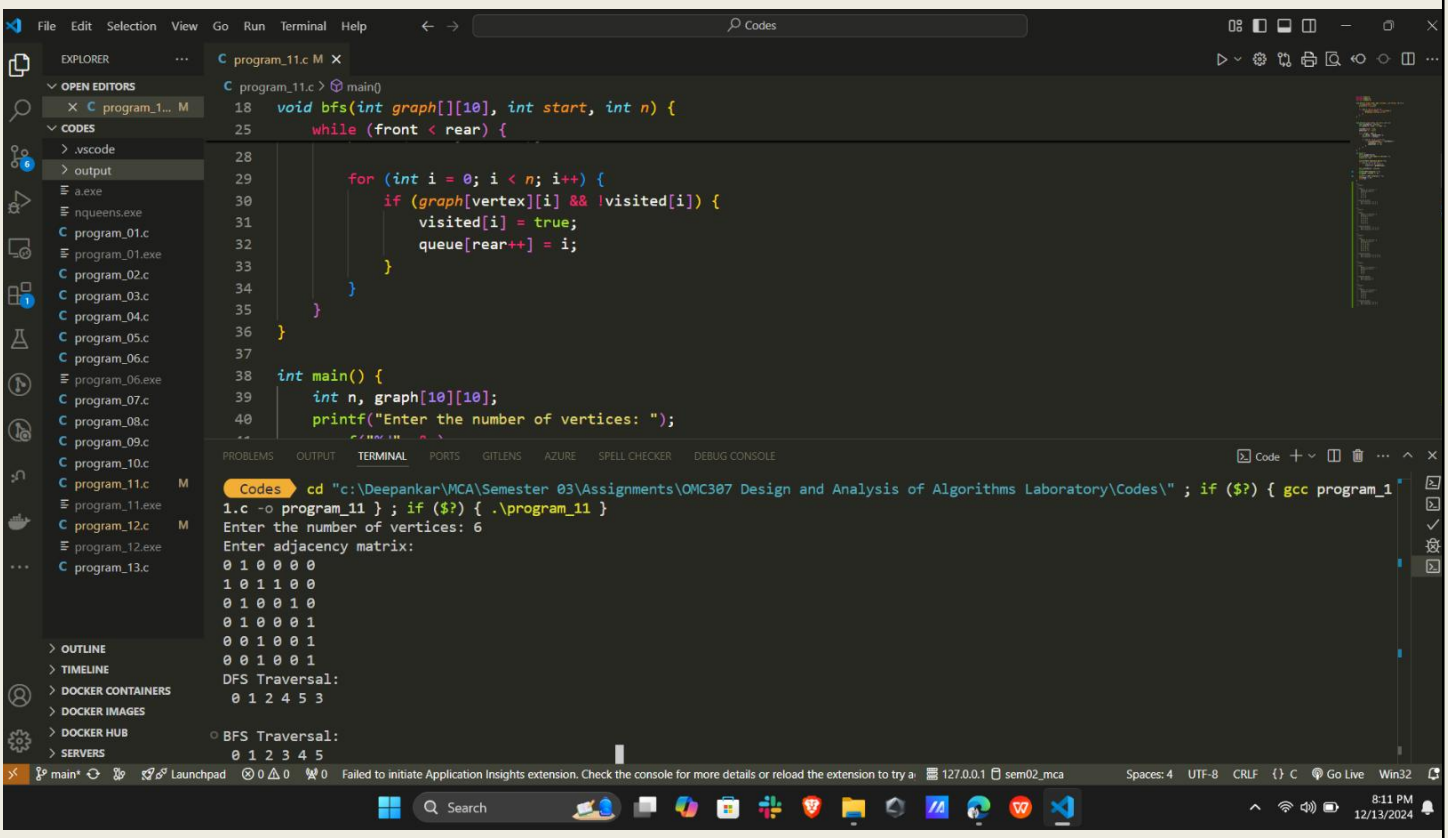
Problem 11 : Graph Traversal : BFS & DFS

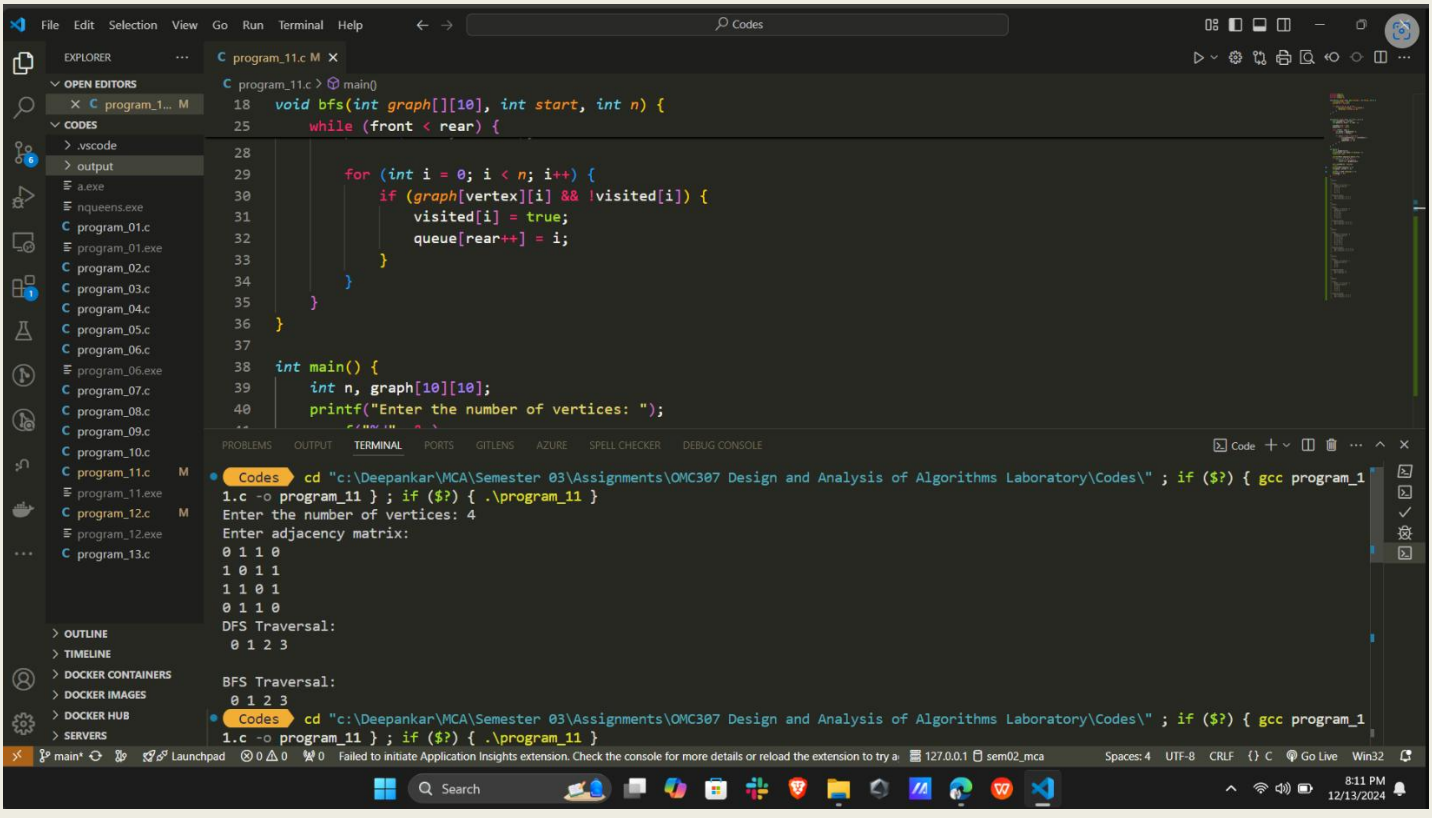
```
void dfs(int graph[X][X], bool visited[], int vertex, int n) {
    visited[vertex] = true; printf("%d", vertex);
for (int(i=0); i<n; i++)
        if(graph[vertex][i] && !visited[i])
            dfs(graph, visited, i, n);
}

void bfs(int graph[V][X], int start, int n){
    bool visited[X] = {false}; int queue[x], front=0, rear=0;
    visited[start] = true; queue[rear++] = start;
    while (front<rear){
        int vertex = queue[front++];
        printf("%d", vertex);
        for(int i=0; i<n; i++) {
            if(graph[vertex][i] && !visited[i]){
                visited[i] = true;
                queue[rear++] = i;
            }
        }
    }
}
```

Output



Output


File Edit Selection View Go Run Terminal Help
Codes
EXPLORER
program_11.c M
program_11.c > main()
void bfs(int graph[][10], int start, int n) {
while (front < rear) {
for (int i = 0; i < n; i++) {
if (graph[vertex][i] && !visited[i]) {
visited[i] = true;
queue[rear++] = i;
}
}
}
}
int main() {
int n, graph[10][10];
printf("Enter the number of vertices: ");
PROBLEMS OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER DEBUG CONSOLE
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
1.c -o program_11 } ; if ($?) { .\program_11 }
Enter the number of vertices: 6
Enter adjacency matrix:
0 1 0 0 0 0
1 0 1 1 0 0
0 1 0 0 1 0
0 1 0 0 0 1
0 0 1 0 0 1
0 0 1 0 0 1
DFS Traversal:
0 1 2 4 5 3
BFS Traversal:
0 1 2 3 4 5
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
SERVERS
Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a
Spaces: 4 UTF-8 CRLF C Go Live Win32
8:11 PM
12/13/2024


File Edit Selection View Go Run Terminal Help
Codes
EXPLORER
program_11.c M
program_11.c > main()
void bfs(int graph[][10], int start, int n) {
while (front < rear) {
for (int i = 0; i < n; i++) {
if (graph[vertex][i] && !visited[i]) {
visited[i] = true;
queue[rear++] = i;
}
}
}
}
int main() {
int n, graph[10][10];
printf("Enter the number of vertices: ");
PROBLEMS OUTPUT TERMINAL PORTS GITLENS AZURE SPELL CHECKER DEBUG CONSOLE
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
1.c -o program_11 } ; if ($?) { .\program_11 }
Enter the number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 1 1
1 1 0 1
0 1 1 0
DFS Traversal:
0 1 2 3
BFS Traversal:
0 1 2 3
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
1.c -o program_11 } ; if ($?) { .\program_11 }
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
DOCKER HUB
SERVERS
Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a
Spaces: 4 UTF-8 CRLF C Go Live Win32
8:11 PM
12/13/2024

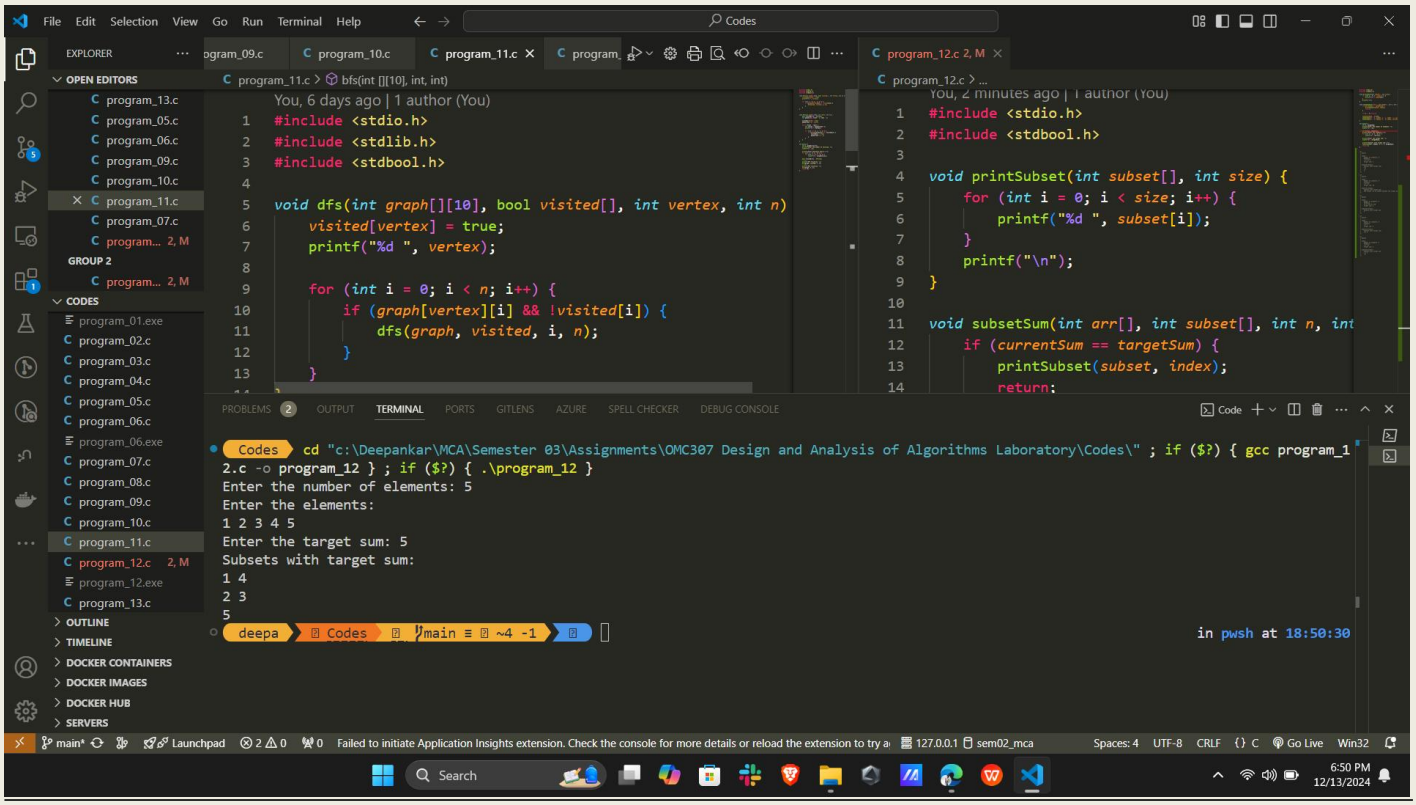
11. Write a C program to find a subset (S = {s_1, s_2, …., s_m}) of a set (U) such that the sum of the elements of the subset (S) is equal to a given integer (d).

Program 12: Subset Sum

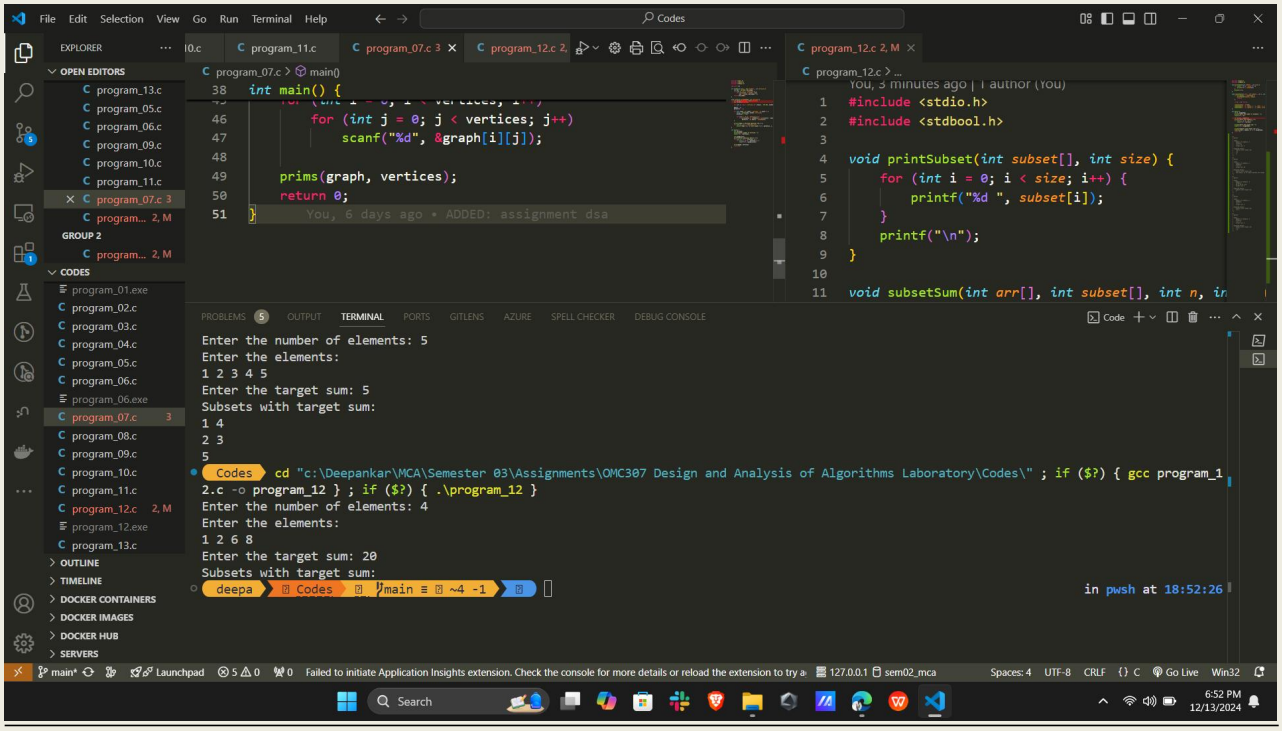
Name : Deepankar Sharma
Course : MCA

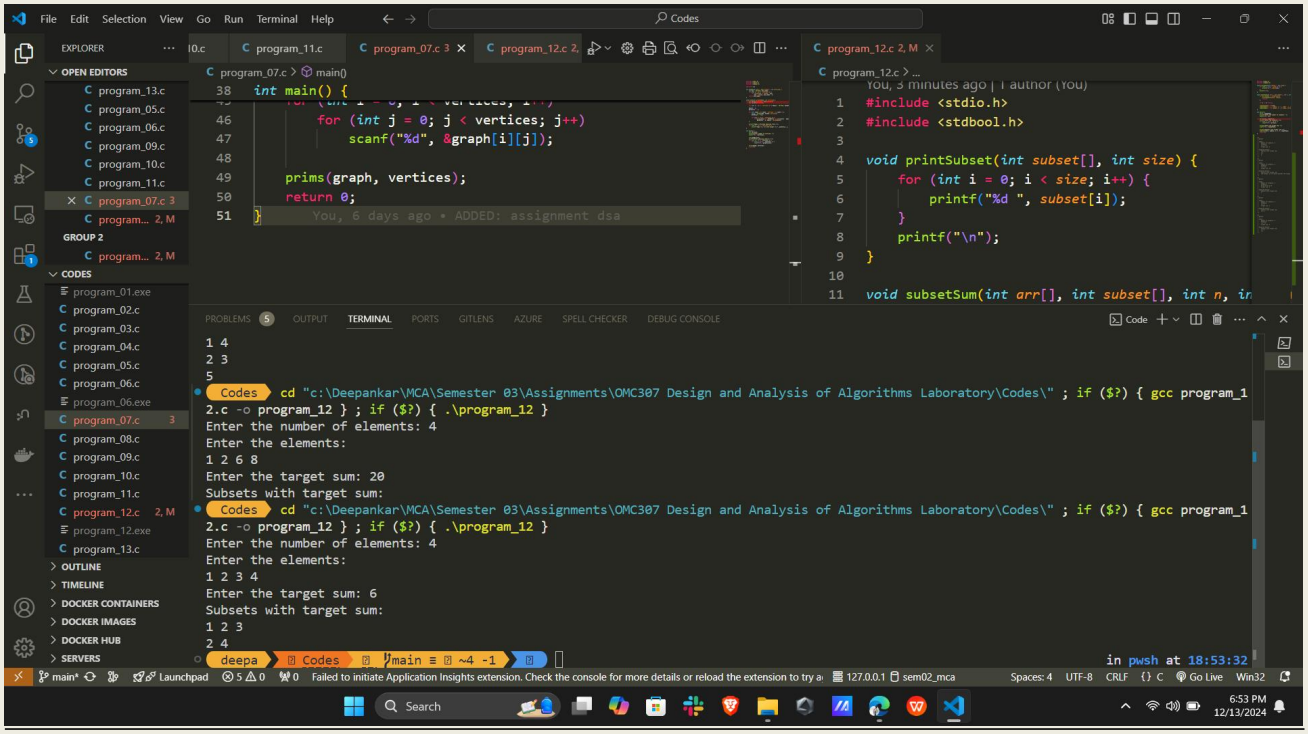
```
void subsetSum(int arr[], int subset[], int n, int index,
               int currentSum, int targetSum) {
    if (currentSum == targetSum) {
        for (int i=0; i < index ; i++) printf("%d ", subset[i]);
        return ;
    }
    if (n==0) return;
    subset[index] = arr[0];
    subsetSum(arr+1, subset, n-1, index+1, currentSum+arr[0],
              targetSum);
    subsetSum(arr+1, subset, n-1, index, currentSum, targetSum);
}
```

output



output


int main() {
for (int j = 0; j < vertices; j++)
scanf("%d", &graph[i][j]);
prims(graph, vertices);
return 0;
}
#include <stdio.h>
#include <stdbool.h>
void printSubset(int subset[], int size) {
for (int i = 0; i < size; i++) {
printf("%d ", subset[i]);
}
printf("\n");
}
void subsetSum(int arr[], int subset[], int n, in
Enter the number of elements: 5
Enter the elements:
1 2 3 4 5
Enter the target sum: 5
Subsets with target sum:
1 4
2 3
5
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
2.c -o program_12 } ; if ($?) { .\program_12 }
Enter the number of elements: 4
Enter the elements:
1 2 6 8
Enter the target sum: 20
Subsets with target sum:
deepa Codes main ≡ ~4 -1
in pwsh at 18:52:26

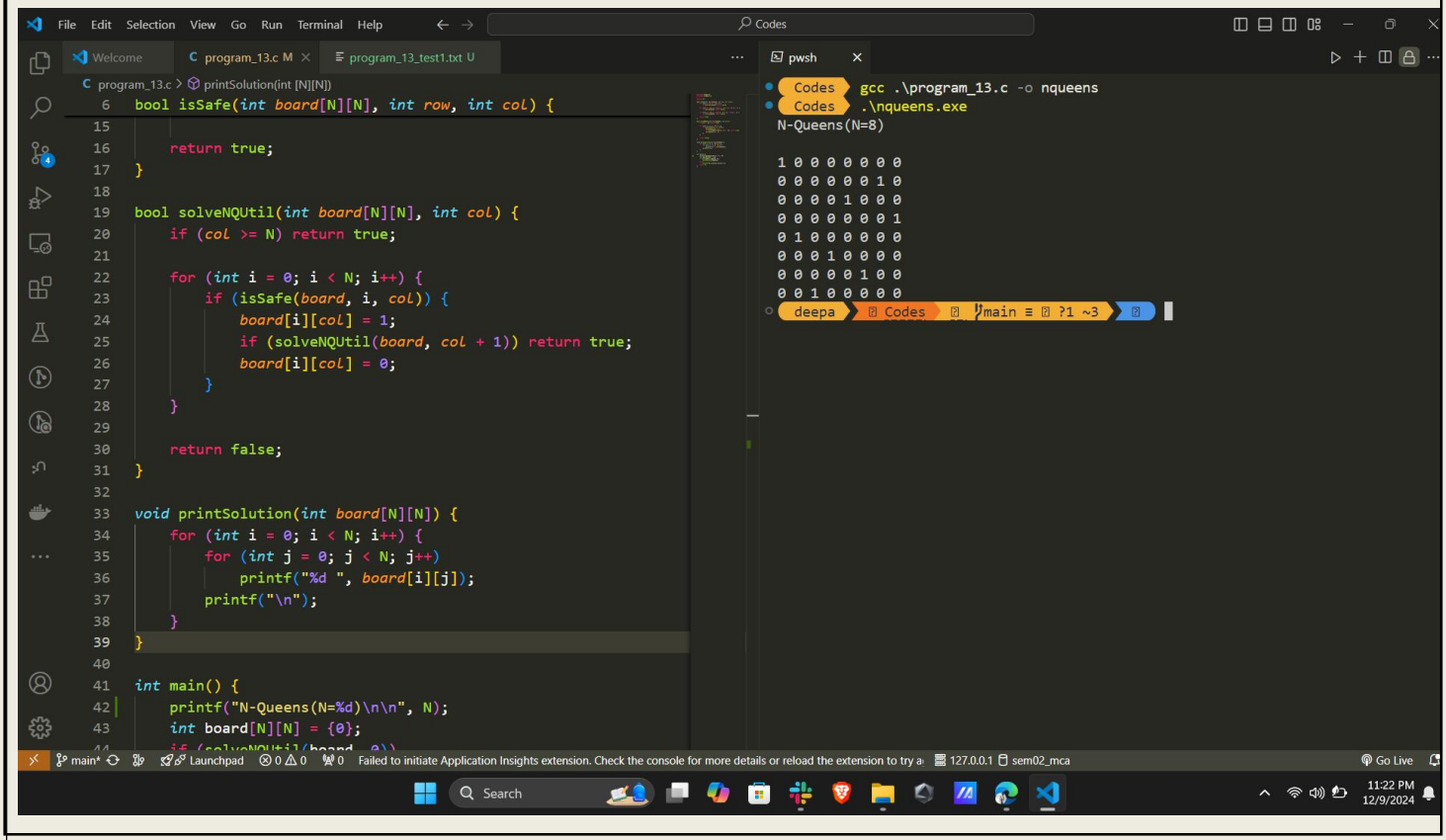

int main() {
for (int j = 0; j < vertices; j++)
scanf("%d", &graph[i][j]);
prims(graph, vertices);
return 0;
}
#include <stdio.h>
#include <stdbool.h>
void printSubset(int subset[], int size) {
for (int i = 0; i < size; i++) {
printf("%d ", subset[i]);
}
printf("\n");
}
void subsetSum(int arr[], int subset[], int n, in
1 4
2 3
5
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
2.c -o program_12 } ; if ($?) { .\program_12 }
Enter the number of elements: 4
Enter the elements:
1 2 6 8
Enter the target sum: 20
Subsets with target sum:
Codes cd "c:\Deepankar\MCA\Semester 03\Assignments\OMC307 Design and Analysis of Algorithms Laboratory\Codes\" ; if ($?) { gcc program_1
2.c -o program_12 } ; if ($?) { .\program_12 }
Enter the number of elements: 4
Enter the elements:
1 2 3 4
Enter the target sum: 6
Subsets with target sum:
1 2 3
2 4
deepa Codes main ≡ ~4 -1
in pwsh at 18:53:32

12. Write a C program to implement n-queen's problem using backtracking.

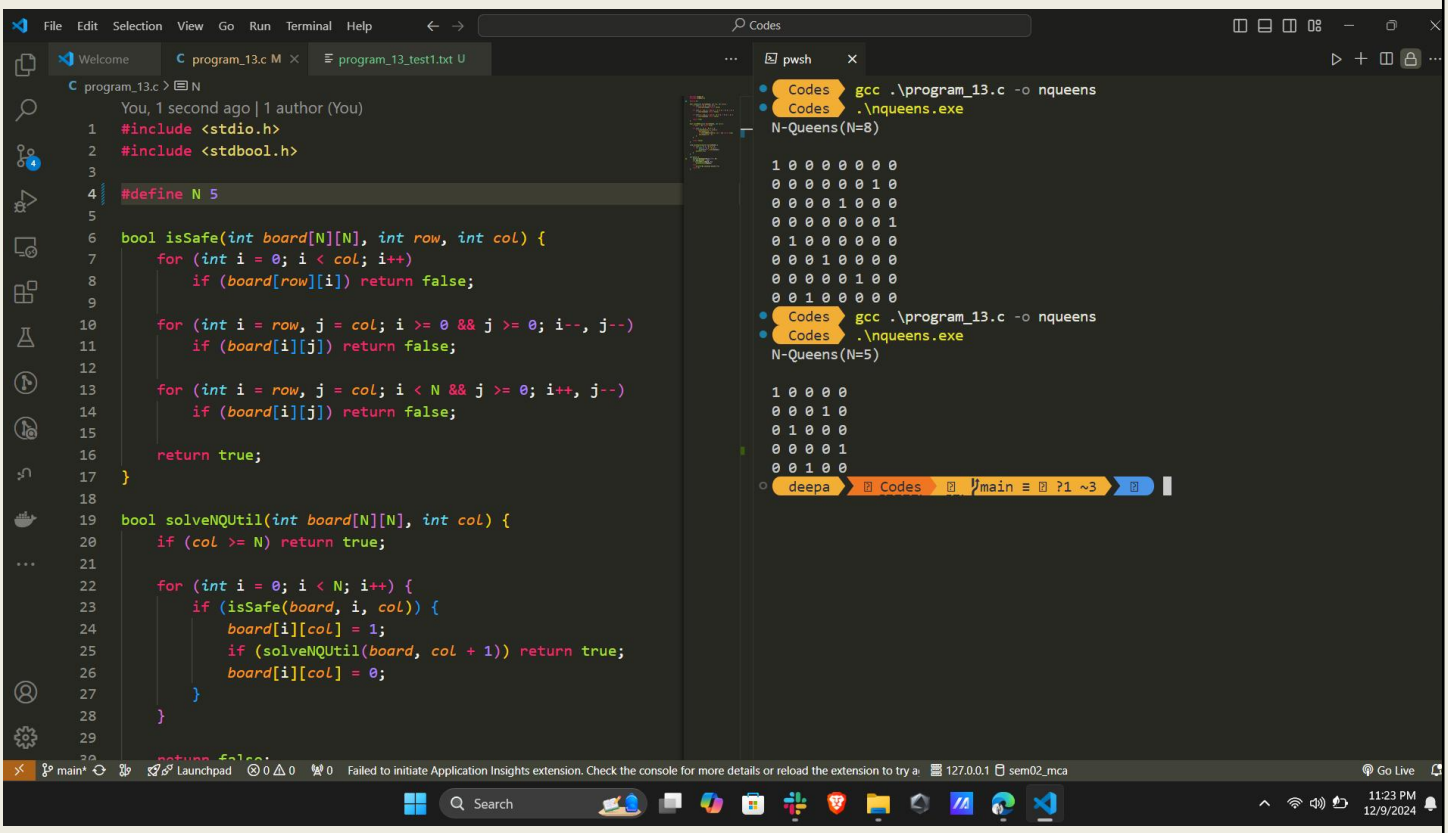
Program 13 : N Queens using Backtracking

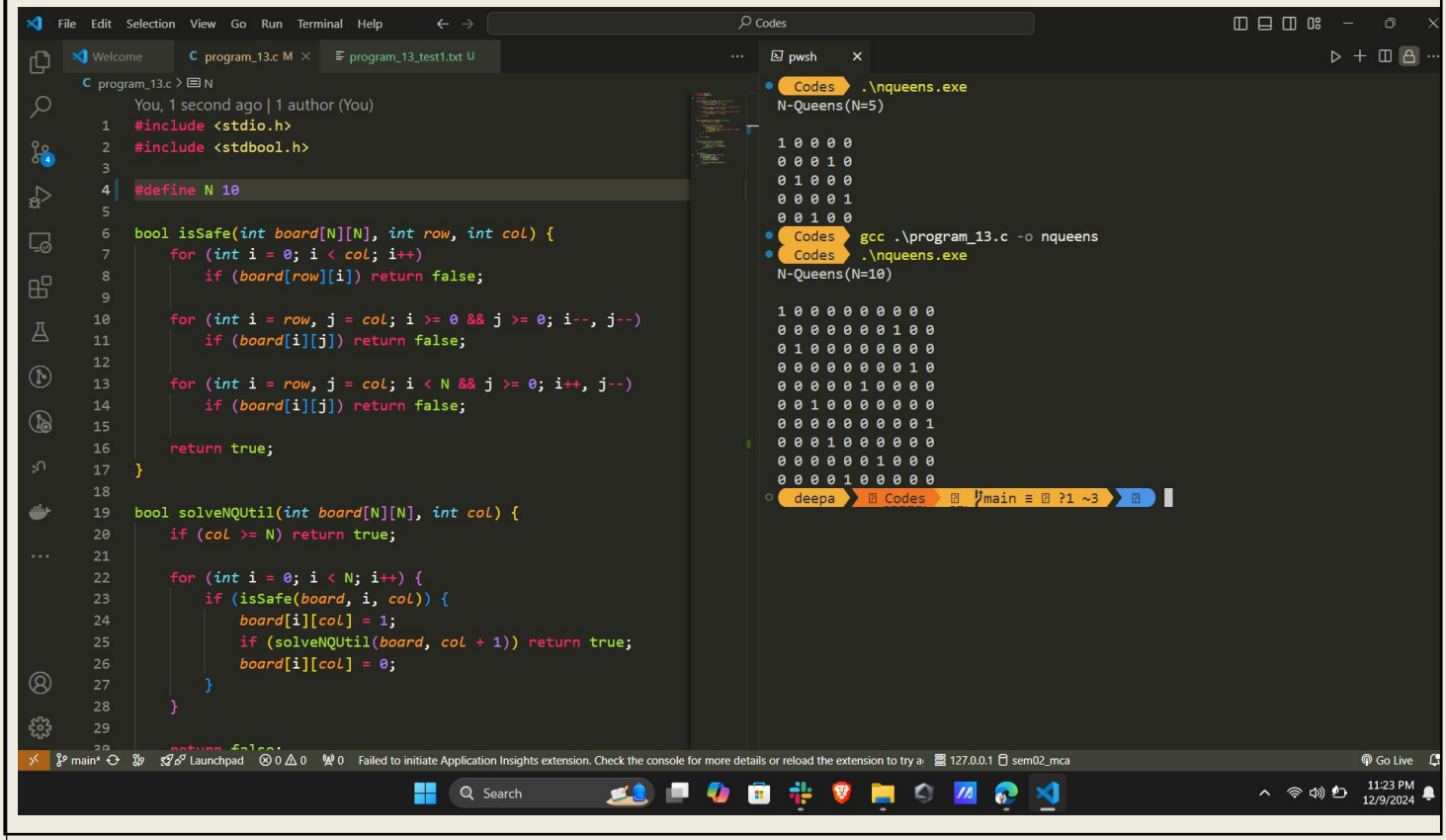
```
#define N 8
bool isSafe (int board [N][N], int row , int col) {
    for ( int i = 0 ; i< col ;i++) if(board [row] [i]) return false ;
    for (int i = row, j = col ; i >=0 && j >=0 ; i-- , j --)
        if (board [i][j]) return false;
    for (int i = row, j=col ; i<N && j >=0 ; i++, j--)
        if (board [i] [j]) return false ;

    return true ;
}
bool solveNQ (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 solveNQ (board , col + 1) return true;
            board [i][col] = 0;
        }
    }
    return false
}
```

Output



Output


File Edit Selection View Go Run Terminal Help
Codes
Welcome
program_13.c M
program_13_test1.txt U
pwsh
You, 1 second ago | 1 author (Yes)
#include <stdio.h>
#include <stdbool.h>
#define N 5
bool isSafe(int board[N][N], int row, int col) {
for (int i = 0; i < col; i++)
if (board[row][i]) return false;
for (int i = row, j = col; i >= 0 && j >= 0; i--, j--)
if (board[i][j]) return false;
for (int i = row, j = col; i < N && j >= 0; 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;
}
}
Codes gcc .\program_13.c -o nqueens
Codes .\nqueens.exe
N-Queens(N=8)
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
Codes gcc .\program_13.c -o nqueens
Codes .\nqueens.exe
N-Queens(N=5)
1 0 0 0 0
0 0 0 1 0
0 1 0 0 0
0 0 0 0 1
0 0 1 0 0
deepa Codes main ?1 ~3
main* Launchpad 0 0 0 Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a 127.0.0.1 sem02_mca Go Live
Search
11:23 PM
12/9/2024


File Edit Selection View Go Run Terminal Help
Codes
Welcome
program_13.c M
program_13_test1.txt U
pwsh
You, 1 second ago | 1 author (Yes)
#include <stdio.h>
#include <stdbool.h>
#define N 10
bool isSafe(int board[N][N], int row, int col) {
for (int i = 0; i < col; i++)
if (board[row][i]) return false;
for (int i = row, j = col; i >= 0 && j >= 0; i--, j--)
if (board[i][j]) return false;
for (int i = row, j = col; i < N && j >= 0; 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;
}
}
Codes .\nqueens.exe
N-Queens(N=5)
1 0 0 0 0
0 0 0 1 0
0 1 0 0 0
0 0 0 0 1
0 0 1 0 0
Codes gcc .\program_13.c -o nqueens
Codes .\nqueens.exe
N-Queens(N=10)
1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1
0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
deepa Codes main ?1 ~3
main* Launchpad 0 0 0 Failed to initiate Application Insights extension. Check the console for more details or reload the extension to try a 127.0.0.1 sem02_mca Go Live
Search
11:23 PM
12/9/2024