**Programs on Backtracking N queens and Sum of Subsets Problem**

#include <stdio.h>

int n;

int check(int i,int j,int a[n][n]){

for(int r=0;r<n;++r)

if(a[i][r]==1) return 0;

for(int r=0;r<n;++r)

if(a[r][j]==1) return 0;

int x=i,y=j;

while(x<n && y<n){

if(a[x][y]==1) return 0;++x;++y;

}

x=i;y=j;

while(x>=0 && y>=0){

if(a[x][y]==1) return 0;--x;--y;}

x=i;y=j;

while(x>=0 && y<n){

if(a[x][y]==1) return 0;--x;++y;}

x=i;y=j;

while(x<n && y>=0){

if(a[x][y]==1) return 0;++x;--y;}

return 1;

}

void print(int a[n][n]){

for(int i=0;i<n;++i){

for(int r=0;r<n;++r)

printf("%d ",a[i][r]);

printf("\n");

}

printf("\n\n");

}

void fn(int i,int a[n][n]){

if(i==n){

print(a);return;

}

for(int r=0;r<n;++r)

if(check(i,r,a)==1){

a[i][r]=1;

fn(i+1,a);

a[i][r]=0;

}

}

int main() {

printf("Enter the Number of queens : ");

scanf("%d",&n);

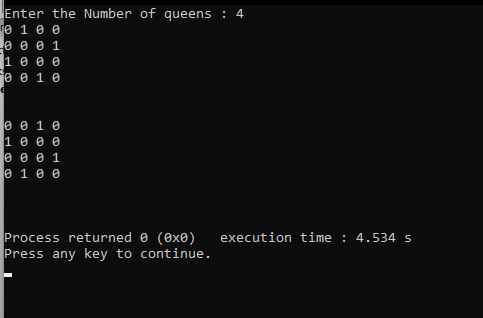
int a[n][n];

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

for(int r=0;r<n;++r) a[i][r]=0;

fn(0,a);

}



**Dynamic Programming Warshall Algorithm.**

#include <stdio.h>

#define N 4

void printMatrix(int matrix[N][N]) {

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

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

printf("%d ", matrix[i][j]);

}

printf("\n");

}

}

void warshall(int graph[N][N]) {

for (int k = 0; k < N; k++) {

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

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

if (graph[i][k] && graph[k][j]) {

graph[i][j] = 1;

}

}

}

}

}

int main() {

int graph[N][N];

printf("Enter the adjacency matrix (0 or 1):\n");

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

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

scanf("%d", &graph[i][j]);

}

}

printf("\nOriginal Adjacency Matrix:\n");

printMatrix(graph);

warshall(graph);

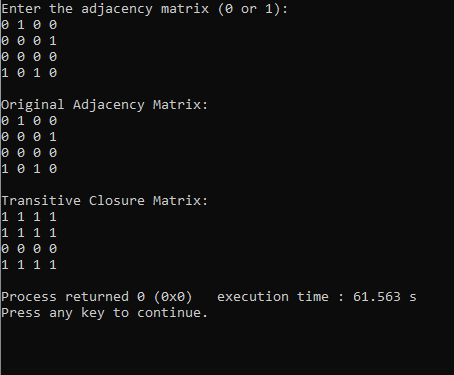
printf("\nTransitive Closure Matrix:\n");

printMatrix(graph);

return 0;

}

Output:



**Sum of subset program:**

**#include <stdio.h>**

**#include <stdbool.h>**

**bool isSubsetSum(int arr[], int n, int sum) {**

**bool dp[n + 1][sum + 1];**

**for (int i = 0; i <= n; i++) {**

**dp[i][0] = true;**

**}**

**for (int i = 1; i <= n; i++) {**

**for (int j = 1; j <= sum; j++) {**

**if (j < arr[i - 1]) {**

**dp[i][j] = dp[i - 1][j];**

**} else {**

**dp[i][j] = dp[i - 1][j] || dp[i - 1][j - arr[i - 1]];**

**}**

**}**

**}**

**return dp[n][sum];**

**}**

**int main() {**

**int n, sum;**

**printf("Enter the number of elements: ");**

**scanf("%d", &n);**

**int arr[n];**

**printf("Enter the elements of the array: ");**

**for (int i = 0; i < n; i++) {**

**scanf("%d", &arr[i]);**

**}**

**printf("Enter the target sum: ");**

**scanf("%d", &sum);**

**if (isSubsetSum(arr, n, sum)) {**

**printf("True\n");**

**} else {**

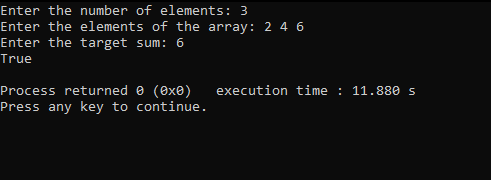
**printf("False\n");**

**}**

**return 0;**

**}**

**Output:**

****