DFS

```
#include <stdio.h>
#include <stdlib.h>
int a[20][20], reach[20], n, trav[20], c=0;
void dfs (int v)
{
  int i;
  trav[c++] = v;
  reach[v] = 1;
  for (i=1; i<=n; i++)
    if (a[v][i] && !reach[i])
      dfs (i);
}
void main()
{
  int i, j, count = 0;
  printf ("\n Enter 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;
  }
  printf ("\n Enter the adjecency matrix: \n");
  for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
      scanf ("%d", &a[i][j]);
  start = clock();
  dfs (1);
  end = clock();
```

```
printf("\n");

for( i=0 ; i<=n ; i++ )
 {
   if (reach [i])
     count++;
}
 if( count ==n )
 printf ("\n Graph is connected\n");
else
printf ("\n Graph is not connected \n");
 for (i=0; i<n; i++)
    printf ("%d\t", taav [i]);
    printf ("\n time required is %.8f\n", ((double)
                          (end - stat) /CLOCKS_PER_SEC));
 }
```

Date
Page

TOH:

```
#include <stdio.h>
#include <time.h>
void towerofHanoi (int n, char from_rod, char to_rod,
                   char aux_rod)
{
if(n==1)
    printf ("Move disc from %c to %c \n", from_rod, to rod);
    return;
}
towerofHanoi (n-1, from_rod, aux_rod, to_rod);
printf ("Move disk from %c to %c \n", from_rod, to_rod);
towerofHanoi (n-1, aux_rod, to_rod, from_rod);
}
int main (int argc, const char* argv[]){
        int n;
    clock_t start, end;
    for(;;){
    printf ("Enter the number of pegs in TOH : ");
      scanf ("%d", &n);
      start = clock();
    towerOfHanoi (n, 'a', 'c', 'b');
  end = clock();
printf ("\n time required is %.8f \n", ((double)(end-start)/
                                        CLOCKS_PER_SEC));}
    return 0;
}
```