JOHNSON TROTTED PROGRAM :-

```c
#include <stdio.h>
#include <conio.h>

int NN, i, count=0;
int p[100], pi[100];      /* The permutation and its inverse */
int dir[100];             /* The directions of each element  */

void PrintPerm()
{
   int i;
   /* uncomment if you want to print the index of each perm  */
   /*
   count = count + 1;
   printf( "[%8d] ", count );
   */
   for (i=1; i <= NN; ++i)
      printf( "%d", p[i] );
} /* PrintPerm */;



void PrintTrans( int x, int y )
{
   printf( " (%d %d)", x, y );
   printf( "\n" );
} /* PrintTrans */;
```

```
void Move( int x, int d )
{
    int z;
    PrintTrans( pi[x], pi[x]+d );
    z = p[pi[x]+d];
    p[pi[x]] = z;
    p[pi[x]+d] = x;
    pi[z] = pi[x];
    pi[x] = pi[x]+d;
} /* Move */;



void Perm ( int n )
{
    int i;
    if (n > NN)
        PrintPerm();
    else
    {
        Perm( n+1 );
        for (i=1; i<=n-1; ++i)
        {
            Move( n, dir[n] );
            Perm( n+1 );
        }
```

```
    {
        Perm( n+1 );
        for (i=1; i<=n-1; ++i)
        {
            Move( n, dir[n] );
            Perm( n+1 );
        }
        dir[n] = -dir[n];
    } // else
} /* of Perm */;

void main ()
{
    printf( "Enter n: " );
    scanf( "%d", &NN );
    printf( "\n" );
    for (i=1; i<=NN; ++i)
    {
        dir[i] = -1; p[i] = i;
        pi[i] = i;
    }
    Perm ( 1 );
    printf( "\n" );
    getch();
} // main()
```

```
Enter n: 4

1234 (4 3)
1243 (3 2)
1423 (2 1)
4123 (4 3)
4132 (1 2)
1432 (2 3)
1342 (3 4)
1324 (2 1)
3124 (4 3)
3142 (3 2)
3412 (2 1)
4312 (4 3)
4321 (1 2)
3421 (2 3)
3241 (3 4)
3214 (1 2)
2314 (4 3)
2341 (3 2)
2431 (2 1)
4231 (3 4)
4213 (1 2)
2413 (2 3)
2143 (3 4)
2134
```