ANAGHA ACHARYA

1BM19CS224

Implement Johnson Trotter algorithm to generate permutations.

#include <stdio.h>

#include<time.h>

int N, i;

int p[1000], q[1000];

int direct[1000];

void Move(int x, int d)

{

int z;

printf("\n");

z = p[q[x]+d];

p[q[x]] = z;

p[q[x]+d] = x;

q[z] = q[x];

q[x] = q[x]+d;

}

void Permutation(int n)

{

int i;

if (n > N){

int i;

for (i=1; i <= N; i++)

printf("%d", p[i]);

}

else

{

Permutation( n+1 );

for (i=1; i<=n-1; ++i)

{

Move(n,direct[n]);

Permutation(n+1);

}

direct[n] = -direct[n];

}

}

void main ()

{

clock\_t start,end;

double time;

printf("Enter the value of N:");

scanf("%d", &N);

printf("\n");

for (i=1; i<=N; ++i)

{

direct[i] = -1; p[i] = i;

q[i] = i;

}

printf("The permutations generated are:\n");

start = clock();

Permutation(1);

end = clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken : %lf\n",time);

}

