**Johnson Trotter**

**ADA LAB - 11/07/2023**

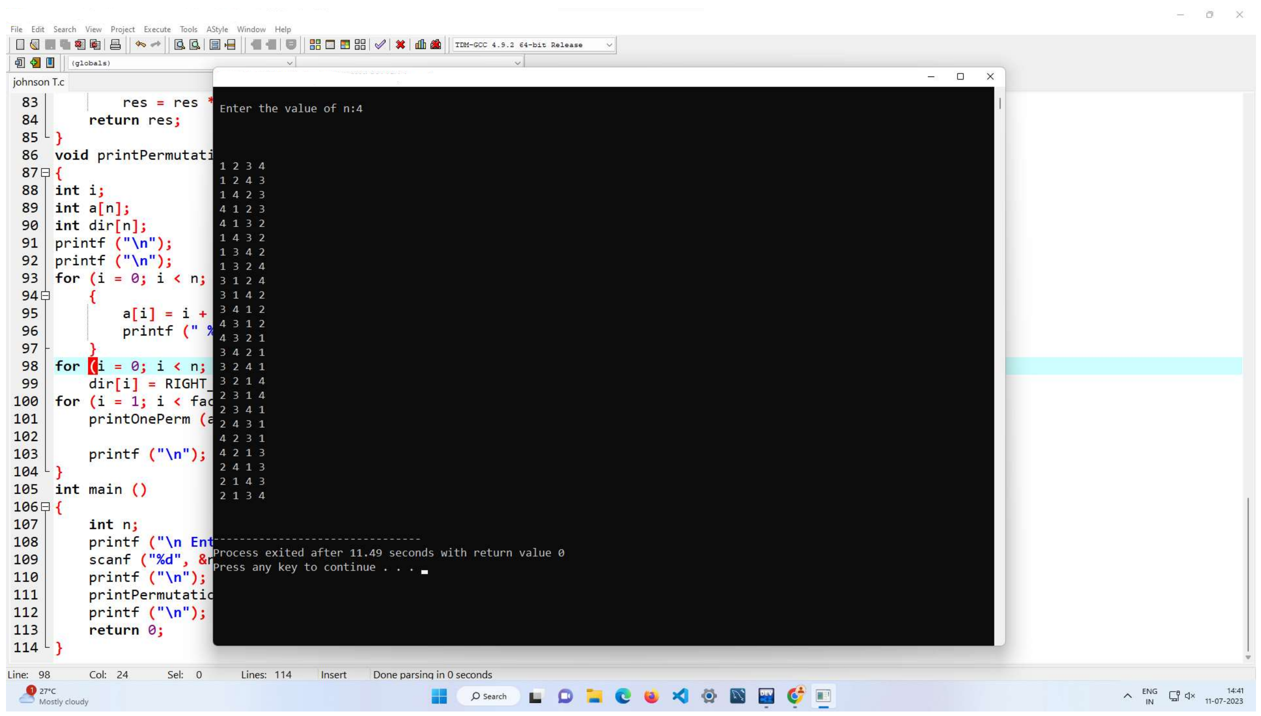
```
#include<stdio.h>
#include<conio.h>

int LEFT_TO_RIGHT = 1;
int RIGHT_TO_LEFT = 0;
int searchArr (int a[], int n, int mobile)
{
        int i;
        for (i = 0; i < n; i++)
        if (a[i] == mobile)
        return i + 1;
}

int getMobile (int a[], int dir[], int n)
{
        int i;
        int mobile_prev = 0, mobile = 0;
        for (i = 0; i < n; i++)
    {
                if (dir[a[i] - 1] == RIGHT_TO_LEFT && i != 0)
        {
                if (a[i] > a[i - 1] && a[i] > mobile_prev)
            {
              mobile = a[i];
                   mobile_prev = mobile;
                }
        }
        if (dir[a[i] - 1] == LEFT_TO_RIGHT && i != n - 1)
        {
                if (a[i] > a[i + 1] && a[i] > mobile_prev)
            {
                    mobile = a[i];
                    mobile_prev = mobile;
                }
        }
}
if (mobile == 0 && mobile_prev == 0)
        return 0;
else
        return mobile;
}

int printOnePerm (int a[], int dir[], int n)
{
int i;
int mobile = getMobile (a, dir, n);
int pos = searchArr (a, n, mobile);
if (dir[a[pos - 1] - 1] == RIGHT_TO_LEFT)
   {
                printf ("\n");
                int temp;
                temp = a[pos - 1];
                a[pos - 1] = a[pos - 2];
                a[pos - 2] = temp;
        }
else if (dir[a[pos - 1] - 1] == LEFT_TO_RIGHT)
{
        printf ("\n");
        int temp;
        temp = a[pos];
        a[pos] = a[pos - 1];
        a[pos - 1] =
        temp;
}
for (i = 0; i < n; i++)
{
        if (a[i] > mobile)
        {
                if (dir[a[i] - 1] == LEFT_TO_RIGHT)
                dir[a[i] - 1] = RIGHT_TO_LEFT;
                else if (dir[a[i] - 1] == RIGHT_TO_LEFT)
                        dir[a[i] - 1] = LEFT_TO_RIGHT;
        }
}
for (i = 0; i < n; i++)
        printf (" %d", a[i]);
}
int fact (int n)
{
        int res = 1;
        int i;
        for (i = 1; i <= n; i++)
        res = res * i;
        return res;
}
void printPermutation (int n)
{
int i;
int a[n];
int dir[n];
printf ("\n");
printf ("\n");
for (i = 0; i < n; i++)
   {
                a[i] = i + 1;
                printf (" %d", a[i]);
        }
for (i = 0; i < n; i++)
        dir[i] = RIGHT_TO_LEFT;
for (i = 1; i < fact (n); i++)
        printOnePerm (a, dir, n);
        printf ("\n");
}
int main ()
{
        int n;
        printf ("\n Enter the value of n:");
        scanf ("%d", &n);
        printf ("\n");
        printPermutation (n);
        printf ("\n");
        return 0;
}
```

**OutPut:**


File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
johnson T.c
Line: 98 Col: 24 Sel: 0 Lines: 114 Insert Done parsing in 0 seconds
27°C Mostly cloudy
Search
ENG IN
14:41 11-07-2023

```
        res = res *
    return res;
}
void printPermutati
{
int i;
int a[n];
int dir[n];
printf ("\n");
printf ("\n");
for (i = 0; i < n;
    {
        a[i] = i +
        printf (" %
    }
for (i = 0; i < n;
    dir[i] = RIGHT_
for (i = 1; i < fac
    printOnePerm (a

    printf ("\n");
}
int main ()
{
    int n;
    printf ("\n Ent
    scanf ("%d", &n
    printf ("\n");
    printPermutatio
    printf ("\n");
    return 0;
}
```

```
Enter the value of n:4



1 2 3 4
1 2 4 3
1 4 2 3
4 1 2 3
4 1 3 2
1 4 3 2
1 3 4 2
1 3 2 4
3 1 2 4
3 1 4 2
3 4 1 2
4 3 1 2
4 3 2 1
3 4 2 1
3 2 4 1
3 2 1 4
2 3 1 4
2 3 4 1
2 4 3 1
4 2 3 1
4 2 1 3
2 4 1 3
2 1 4 3
2 1 3 4


--------------------------------
Process exited after 11.49 seconds with return value 0
Press any key to continue . . .
```