

C johnson_trotter.c X

C johnson_trotter.c

```
1  /*Implement Johnson Trotter algorithm to generate permutations.*/
2  #include <stdio.h>
3  int right=1;
4  int left=0;
5
6  int search(int n,int a[n],int mobile)
7  {
8      int p;
9      for (int i=0;i<n;i++)
10     {
11         if (a[i]==mobile)
12         {
13             p=i+1;
14         }
15     }
16     return p;
17 }
18
19 int mobile_element(int n,int a[n],int dir[n])
20 {
21     int prev=0;
22     int mobile=0;
23     for (int i=0;i<n;i++)
24     {
25         if (dir[a[i]-1]==right && i!=0)
26         {
27             if (a[i]>a[i-1] && a[i]>prev)
28             {
29                 mobile=a[i];
30                 prev=mobile;
31             }
32         }
33         if (dir[a[i]-1]==left && i!=(n-1))
34         {
35             if (a[i]>a[i+1] && a[i]>prev)
36             {
37                 mobile=a[i];
38                 prev=mobile;
```

C johnson_trotter.c X

C johnson_trotter.c

```
37         mobile=a[i];
38         prev=mobile;
39     }
40 }
41 }
42 if (mobile==0 && prev==0)
43 {
44     return 0;
45 }
46 return mobile;
47 }
48
49 void printOnePermutation(int n,int a[n],int dir[n])
50 {
51
52     int temp;
53     int mobile=mobile_element(n,a,dir);
54     int pos=search(n,a,mobile);
55     if (dir[a[pos-1]-1]==right)
56     {
57         temp=a[pos-1];
58         a[pos-1]=a[pos-2];
59         a[pos-2]=temp;
60     }
61     else if (dir[a[pos-1]-1]==left)
62     {
63         temp=a[pos];
64         a[pos]=a[pos-1];
65         a[pos-1]=temp;
66     }
67     for (int i=0;i<n;i++)
68     {
69         if (a[i]>mobile)
70         {
71             if (dir[a[i]-1]==left)
72             {
73                 dir[a[i]-1]=right;
74             }
```

C johnson_trotter.c X

C johnson_trotter.c

```
73     dir[a[i]-1]=right;
74 }
75 else if (dir[a[i]-1]==right)
76 {
77     dir[a[i]-1]=left;
78 }
79 }
80 }
81 for (int i=0;i<n;i++)
82 {
83     printf("%d",a[i]);
84 }
85 printf("\n");
86 }
87
88 int fact(int n)
89 {
90     int product=1;
91     for (int i=1;i<=n;i++)
92     {
93         product=product*i;
94     }
95     return product;
96 }
97
98 void printPermutation(int n)
99 {
100     int a[n];
101     int dir[n];
102     for (int i=0;i<n;i++)
103     {
104         a[i]=i+1;
105         printf("%d",a[i]);
106     }
107     printf("\n");
108     for (int i=0;i<n;i++)
109     {
110         dir[i]=right;
111     }
112 }
```

C johnson_trotter.c X

C johnson_trotter.c

```
99  {
100      int a[n];
101      int dir[n];
102      for (int i=0;i<n;i++)
103      {
104          a[i]=i+1;
105          printf("%d",a[i]);
106      }
107      printf("\n");
108      for (int i=0;i<n;i++)
109      {
110          dir[i]=right;
111      }
112      for (int i=1;i<fact(n);i++)
113      {
114          printOnePermutation(n,a,dir);
115      }
116  }
117 }
118
119 int main()
120 {
121     int n;
122     printf("Enter the value of n:\n");
123     scanf("%d",&n);
124     printPermutation(n);
125     return 0;
126 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell



PS C:\Users\muska\OneDrive\Desktop\C programs> gcc johnson_trotter.c

PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe

Enter the value of n:

4

1234

1243

1423

4123

4132

1432

1342

1324

3124

3142

3412

4312

4321

3421

3241

3214

2314

2341

2431

4231

4213

2413

2143

2134

PS C:\Users\muska\OneDrive\Desktop\C programs> |