

PROGRAM 7

Implement Johnson Trotter algorithm to generate permutations

//Code

```
#include<iostream>
```

```
using namespace std;
```

```
int right_dir=1;
```

```
int left_dir=0;
```

```
int fact(int n){
```

```
    if(n==1)
```

```
        return 1;
```

```
    else
```

```
        return n*fact(n-1);
```

```
}
```

```
int mobile_key(int a[],int dir[],int n){
```

```
    int mobile=0;
```

```
    for(int i=0;i<n;i++){
```

```
        if(dir[a[i]-1]==left_dir && i!=0){
```

```
            if(a[i]>a[i-1] && a[i]>mobile)
```

```
                mobile=a[i];
```

```
        }
```

```
        if(dir[a[i]-1]==right_dir && i!=n-1){
```

```
            if(a[i]>a[i+1] && a[i]>mobile)
```

```
                mobile=a[i];
```

```
        }
```

```
    }
```

```
    return mobile;
```

```
}
```

```
int search_mob(int a[],int n,int mobile){
```

```

int pos;
for(int i=0;i<n;i++)
    if(a[i]==mobile)
        pos=i+1;
return pos;
}

```

```

void printPermutation(int a[],int dir[],int n){
    int mobile=mobile_key(a,dir,n);
    int pos=search_mob(a,n,mobile);
    if(dir[a[pos-1]-1]==left_dir)
        swap(a[pos-1],a[pos-2]);
    else if(dir[a[pos-1]-1]==right_dir)
        swap(a[pos],a[pos-1]);
    for(int i=0;i<n;i++){
        if(a[i]>mobile){
            if(dir[a[i]-1]==right_dir)
                dir[a[i]-1]=left_dir;
            else
                dir[a[i]-1]=right_dir;
        }
    }
    for(int i=0;i<n;i++){
        cout<<a[i];
        cout<<endl;
    }
}

```

```

void JohnsonTrotter(int n){
    int a[n],dir[n];
    for(int i=0;i<n;i++){
        a[i]=i+1;
        dir[i]=left_dir;
    }
    cout<<"Permutations: "<<endl;
    for(int i=0;i<n;i++){
        cout<<a[i];
    }
}

```

```
    cout<<endl;
    for(int i=1;i<fact(n);i++)
        printPermutation(a,dir,n);
}
```

```
int main(){
    int n;
    cout<<"Enter n:";
    cin>>n;
    JohnsonTrotter(n);
}
```

//Output

```
❯ clang++-7 -pthread -std=c++17 -o main main.cpp
❯ ./main
Enter n:4
Permutations:
1234
1243
1423
4123
4132
1432
1342
1324
3124
3142
3412
4312
4321
3421
3241
3214
2314
2341
2431
4231
4213
2413
2143
2134
❯
```