```
==================================================================
                         1
==================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
1 4 3 5 2

--------------
                n=5, i=n-2=3
                here, v[3]=5, v[3+1]=2
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
stage B
stage C
                k=n-1=4
checking v[i]>v[k] or not.
--------------  v[2]=3, v[4]=2
---++++-------  v[2]=3 > v[4]=2
decrementing k, k=3

--------------  k=3
stage D
n=5, i=2, k=3
---Always swapped --- before swap, v[i=2]=3, v[k=3]=5
---------------------- swaped v[i=2]=5, v[k=3]=3
------++++++++ The current array is:
1 4 5 3 2
assigning: k=0
checking j < (n+i)/2+1 or not.
n=5, i=2, j=3, k=0
------++++++++++  j=3, (n+i)/2+1=4, k=0
------++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[3]=3, v[4]=2
-----+++++++++ swaped v[3]=2, v[4]=3
------++++++++ The current array is:
1 4 5 2 3
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
1 4 5 2 3
Press any key to continue . . .
==================================================================
                         2
==================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
5 4 1 3 2

--------------
                n=5, i=n-2=3
                here, v[3]=3, v[3+1]=2
```

```
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
stage B
stage C
                k=n-1=4
checking v[i]>v[k] or not.
--------------  v[2]=1, v[4]=2

--------------  k=4
stage D
n=5, i=2, k=4
---Always swapped --- before swap, v[i=2]=1, v[k=4]=2
---------------------- swaped v[i=2]=2, v[k=4]=1
------++++++++ The current array is:
5 4 2 3 1
assigning: k=0
checking j < (n+i)/2+1 or not.
n=5, i=2, j=3, k=0
------++++++++++  j=3, (n+i)/2+1=4, k=0
------++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[3]=3, v[4]=1
-----+++++++++ swaped v[3]=1, v[4]=3
------++++++++ The current array is:
5 4 2 1 3
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
5 4 2 1 3
Press any key to continue . . .
================================================================
                              3
================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
4 3 2 5 1

--------------
                n=5, i=n-2=3
                here, v[3]=5, v[3+1]=1
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
stage B
stage C
                k=n-1=4
checking v[i]>v[k] or not.
--------------  v[2]=2, v[4]=1
---++++-------  v[2]=2 > v[4]=1
decrementing k, k=3

--------------  k=3
stage D
n=5, i=2, k=3
```

```
---Always swapped --- before swap, v[i=2]=2, v[k=3]=5
---------------------- swaped v[i=2]=5, v[k=3]=2
------++++++++ The current array is:
4 3 5 2 1
assigning: k=0
checking j < (n+i)/2+1 or not.
n=5, i=2, j=3, k=0
------++++++++++  j=3, (n+i)/2+1=4, k=0
------++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[3]=2, v[4]=1
-----+++++++++ swaped v[3]=1, v[4]=2
------++++++++ The current array is:
4 3 5 1 2
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
4 3 5 1 2
Press any key to continue . . .
======================================================================
                             4
======================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
1 4 5 3 2

--------------
               n=5, i=n-2=3
               here, v[3]=3, v[3+1]=2
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
(2>=0) && (v[2]>v[2+1]) met, decrementing i
i decremented, i=1
stage B
stage C
               k=n-1=4
checking v[i]>v[k] or not.
--------------  v[1]=4, v[4]=2
---++++-------  v[1]=4 > v[4]=2
decrementing k, k=3
---++++-------  v[1]=4 > v[3]=3
decrementing k, k=2

--------------  k=2
stage D
n=5, i=1, k=2
---Always swapped --- before swap, v[i=1]=4, v[k=2]=5
---------------------- swaped v[i=1]=5, v[k=2]=4
------++++++++ The current array is:
1 5 4 3 2
assigning: k=0
checking j < (n+i)/2+1 or not.
n=5, i=1, j=2, k=0
------++++++++++  j=2, (n+i)/2+1=4, k=0
```

```
------++++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[2]=4, v[4]=2
-----+++++++++ swaped v[2]=2, v[4]=4
------++++++++ The current array is:
1 5 2 3 4
incrementing j and k
checking j < (n+i)/2+1 or not.
n=5, i=1, j=3, k=1
------++++++++++++  j=3, (n+i)/2+1=4, k=1
------++++++++++++  n-k-1=3
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[3]=3, v[3]=3
-----+++++++++ swaped v[3]=3, v[3]=3
------++++++++ The current array is:
1 5 2 3 4
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
1 5 2 3 4
Press any key to continue . . .
================================================================
                                5
================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
1 5 4 3 2

--------------
               n=5, i=n-2=3
               here, v[3]=3, v[3+1]=2
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
(2>=0) && (v[2]>v[2+1]) met, decrementing i
i decremented, i=1
(1>=0) && (v[1]>v[1+1]) met, decrementing i
i decremented, i=0
stage B
stage C
               k=n-1=4
checking v[i]>v[k] or not.
--------------  v[0]=1, v[4]=2

--------------  k=4
stage D
n=5, i=0, k=4
---Always swapped --- before swap, v[i=0]=1, v[k=4]=2
-------------------- swaped v[i=0]=2, v[k=4]=1
------++++++++ The current array is:
2 5 4 3 1
assigning: k=0
checking j < (n+i)/2+1 or not.
n=5, i=0, j=1, k=0
------++++++++++++  j=1, (n+i)/2+1=3, k=0
```

```
------++++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[1]=5, v[4]=1
-----+++++++++ swaped v[1]=1, v[4]=5
------++++++++ The current array is:
2 1 4 3 5
incrementing j and k
checking j < (n+i)/2+1 or not.
n=5, i=0, j=2, k=1
------++++++++++++  j=2, (n+i)/2+1=3, k=1
------++++++++++++  n-k-1=3
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[2]=4, v[3]=3
-----+++++++++ swaped v[2]=3, v[3]=4
------++++++++ The current array is:
2 1 3 4 5
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
2 1 3 4 5
Press any key to continue . . .
================================================================
                                6
================================================================
Know the next permutation.
how long your array is (no. of elements):
5
input the elements with spaces:
4 3 5 2 1

--------------
               n=5, i=n-2=3
               here, v[3]=2, v[3+1]=1
stage A
(3>=0) && (v[3]>v[3+1]) met, decrementing i
i decremented, i=2
(2>=0) && (v[2]>v[2+1]) met, decrementing i
i decremented, i=1
stage B
stage C
               k=n-1=4
checking v[i]>v[k] or not.
--------------  v[1]=3, v[4]=1
---+++++------  v[1]=3 > v[4]=1
decrementing k, k=3
---+++++------  v[1]=3 > v[3]=2
decrementing k, k=2

--------------  k=2
stage D
n=5, i=1, k=2
---Always swapped --- before swap, v[i=1]=3, v[k=2]=5
--------------------- swaped v[i=1]=5, v[k=2]=3
------++++++++ The current array is:
4 5 3 2 1
assigning: k=0
checking j < (n+i)/2+1 or not.
```

```
n=5, i=1, j=2, k=0
------++++++++++++  j=2, (n+i)/2+1=4, k=0
------++++++++++++  n-k-1=4
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[2]=3, v[4]=1
-----+++++++++ swaped v[2]=1, v[4]=3
------++++++++ The current array is:
4 5 1 2 3
incrementing j and k
checking j < (n+i)/2+1 or not.
n=5, i=1, j=3, k=1
------++++++++++++  j=3, (n+i)/2+1=4, k=1
------++++++++++++  n-k-1=3
-----+++++++++ preparing swap v[j] and v[n-k-1]
-----+++++++++ before swap v[3]=2, v[3]=2
-----+++++++++ swaped v[3]=2, v[3]=2
------++++++++ The current array is:
4 5 1 2 3
incrementing j and k
       returning 0
............
returned to main(). Next permutation is:
4 5 1 2 3
Press any key to continue . . .
================================================================
                              7
================================================================
```