

```

1  =====
2                      1
3  =====
4  Know the next permutation.
5  how long your array is (no. of elements):
6  5
7  input the elements with spaces:
8  1 4 3 5 2
9
10 -----
11             n=5, i=n-2=3
12             here, v[3]=5, v[3+1]=2
13 stage A
14 (3>=0) && (v[3]>v[3+1]) met, decrementing i
15 i decremented, i=2
16 stage B
17 stage C
18             k=n-1=4
19 checking v[i]>v[k] or not.
20 ----- v[2]=3, v[4]=2
21 ---++++----- v[2]=3 > v[4]=2
22 decrementing k, k=3
23
24 ----- k=3
25 stage D
26 n=5, i=2, k=3
27 ---Always swapped --- before swap, v[i=2]=3, v[k=3]=5
28 ----- swaped v[i=2]=5, v[k=3]=3
29 -----+++++++ The current array is:
30 1 4 5 3 2
31 assigning: k=0
32 checking j < (n+i)/2+1 or not.
33 n=5, i=2, j=3, k=0
34 -----+++++++ j=3, (n+i)/2+1=4, k=0
35 -----+++++++ n-k-1=4
36 -----+++++++ preparing swap v[j] and v[n-k-1]
37 -----+++++++ before swap v[3]=3, v[4]=2
38 -----+++++++ swaped v[3]=2, v[4]=3
39 -----+++++++ The current array is:
40 1 4 5 2 3
41 incrementing j and k
42         returning 0
43 .....
44 returned to main(). Next permutation is:
45 1 4 5 2 3
46 Press any key to continue . . .
47 =====
48                      2
49 =====
50 Know the next permutation.
51 how long your array is (no. of elements):
52 5
53 input the elements with spaces:
54 5 4 1 3 2
55
56 -----
57             n=5, i=n-2=3
58             here, v[3]=3, v[3+1]=2

```

```

59 stage A
60 (3>=0) && (v[3]>v[3+1]) met, decrementing i
61 i decremented, i=2
62 stage B
63 stage C
64             k=n-1=4
65 checking v[i]>v[k] or not.
66 ----- v[2]=1, v[4]=2
67
68 ----- k=4
69 stage D
70 n=5, i=2, k=4
71 ---Always swapped --- before swap, v[i=2]=1, v[k=4]=2
72 ----- swaped v[i=2]=2, v[k=4]=1
73 -----+++++++ The current array is:
74 5 4 2 3 1
75 assigning: k=0
76 checking j < (n+i)/2+1 or not.
77 n=5, i=2, j=3, k=0
78 -----+++++++ j=3, (n+i)/2+1=4, k=0
79 -----+++++++ n-k-1=4
80 -----+++++++ preparing swap v[j] and v[n-k-1]
81 -----+++++++ before swap v[3]=3, v[4]=1
82 -----+++++++ swaped v[3]=1, v[4]=3
83 -----+++++++ The current array is:
84 5 4 2 1 3
85 incrementing j and k
86         returning 0
87 .....
88 returned to main(). Next permutation is:
89 5 4 2 1 3
90 Press any key to continue . . .
91 =====
92                               3
93 =====
94 Know the next permutation.
95 how long your array is (no. of elements):
96 5
97 input the elements with spaces:
98 4 3 2 5 1
99
100 -----
101             n=5, i=n-2=3
102             here, v[3]=5, v[3+1]=1
103 stage A
104 (3>=0) && (v[3]>v[3+1]) met, decrementing i
105 i decremented, i=2
106 stage B
107 stage C
108             k=n-1=4
109 checking v[i]>v[k] or not.
110 ----- v[2]=2, v[4]=1
111 -----+++++ v[2]=2 > v[4]=1
112 decrementing k, k=3
113
114 ----- k=3
115 stage D
116 n=5, i=2, k=3

```

```

117 ---Always swapped --- before swap, v[i=2]=2, v[k=3]=5
118 -----swapped v[i=2]=5, v[k=3]=2
119 -----+++++++ The current array is:
120 4 3 5 2 1
121 assigning: k=0
122 checking j < (n+i)/2+1 or not.
123 n=5, i=2, j=3, k=0
124 -----+++++++ j=3, (n+i)/2+1=4, k=0
125 -----+++++++ n-k-1=4
126 -----+++++++ preparing swap v[j] and v[n-k-1]
127 -----+++++++ before swap v[3]=2, v[4]=1
128 -----+++++++ swaped v[3]=1, v[4]=2
129 -----+++++++ The current array is:
130 4 3 5 1 2
131 incrementing j and k
132         returning 0
133 .....
134 returned to main(). Next permutation is:
135 4 3 5 1 2
136 Press any key to continue . . .
137 =====
138                                     4
139 =====
140 Know the next permutation.
141 how long your array is (no. of elements):
142 5
143 input the elements with spaces:
144 1 4 5 3 2
145
146 -----
147             n=5, i=n-2=3
148             here, v[3]=3, v[3+1]=2
149 stage A
150 (3>=0) && (v[3]>v[3+1]) met, decrementing i
151 i decremented, i=2
152 (2>=0) && (v[2]>v[2+1]) met, decrementing i
153 i decremented, i=1
154 stage B
155 stage C
156             k=n-1=4
157 checking v[i]>v[k] or not.
158 -----v[1]=4, v[4]=2
159 ---++++----- v[1]=4 > v[4]=2
160 decrementing k, k=3
161 ---++++----- v[1]=4 > v[3]=3
162 decrementing k, k=2
163
164 -----k=2
165 stage D
166 n=5, i=1, k=2
167 ---Always swapped --- before swap, v[i=1]=4, v[k=2]=5
168 -----swaped v[i=1]=5, v[k=2]=4
169 -----+++++++ The current array is:
170 1 5 4 3 2
171 assigning: k=0
172 checking j < (n+i)/2+1 or not.
173 n=5, i=1, j=2, k=0
174 -----+++++++ j=2, (n+i)/2+1=4, k=0

```

```

175 -----+++++++ n-k-1=4
176 -----+++++++ preparing swap v[j] and v[n-k-1]
177 -----+++++++ before swap v[2]=4, v[4]=2
178 -----+++++++ swaped v[2]=2, v[4]=4
179 -----+++++++ The current array is:
180 1 5 2 3 4
181 incrementing j and k
182 checking j < (n+i)/2+1 or not.
183 n=5, i=1, j=3, k=1
184 -----+++++++ j=3, (n+i)/2+1=4, k=1
185 -----+++++++ n-k-1=3
186 -----+++++++ preparing swap v[j] and v[n-k-1]
187 -----+++++++ before swap v[3]=3, v[3]=3
188 -----+++++++ swaped v[3]=3, v[3]=3
189 -----+++++++ The current array is:
190 1 5 2 3 4
191 incrementing j and k
192         returning 0
193 .....
194 returned to main(). Next permutation is:
195 1 5 2 3 4
196 Press any key to continue . . .
197 =====
198                                     5
199 =====
200 Know the next permutation.
201 how long your array is (no. of elements):
202 5
203 input the elements with spaces:
204 1 5 4 3 2
205
206 -----
207             n=5, i=n-2=3
208             here, v[3]=3, v[3+1]=2
209 stage A
210 (3>=0) && (v[3]>v[3+1]) met, decrementing i
211 i decremented, i=2
212 (2>=0) && (v[2]>v[2+1]) met, decrementing i
213 i decremented, i=1
214 (1>=0) && (v[1]>v[1+1]) met, decrementing i
215 i decremented, i=0
216 stage B
217 stage C
218             k=n-1=4
219 checking v[i]>v[k] or not.
220 ----- v[0]=1, v[4]=2
221
222 ----- k=4
223 stage D
224 n=5, i=0, k=4
225 ---Always swaped --- before swap, v[i=0]=1, v[k=4]=2
226 ----- swaped v[i=0]=2, v[k=4]=1
227 -----+++++++ The current array is:
228 2 5 4 3 1
229 assigning: k=0
230 checking j < (n+i)/2+1 or not.
231 n=5, i=0, j=1, k=0
232 -----+++++++ j=1, (n+i)/2+1=3, k=0

```

```

233 -----+++++++ n-k-1=4
234 -----+++++++ preparing swap v[j] and v[n-k-1]
235 -----+++++++ before swap v[1]=5, v[4]=1
236 -----+++++++ swaped v[1]=1, v[4]=5
237 -----+++++++ The current array is:
238 2 1 4 3 5
239 incrementing j and k
240 checking j < (n+i)/2+1 or not.
241 n=5, i=0, j=2, k=1
242 -----+++++++ j=2, (n+i)/2+1=3, k=1
243 -----+++++++ n-k-1=3
244 -----+++++++ preparing swap v[j] and v[n-k-1]
245 -----+++++++ before swap v[2]=4, v[3]=3
246 -----+++++++ swaped v[2]=3, v[3]=4
247 -----+++++++ The current array is:
248 2 1 3 4 5
249 incrementing j and k
250         returning 0
251 .....
252 returned to main(). Next permutation is:
253 2 1 3 4 5
254 Press any key to continue . . .
255 =====
256                                     6
257 =====
258 Know the next permutation.
259 how long your array is (no. of elements):
260 5
261 input the elements with spaces:
262 4 3 5 2 1
263 -----
264
265             n=5, i=n-2=3
266             here, v[3]=2, v[3+1]=1
267 stage A
268 (3>=0) && (v[3]>v[3+1]) met, decrementing i
269 i decremented, i=2
270 (2>=0) && (v[2]>v[2+1]) met, decrementing i
271 i decremented, i=1
272 stage B
273 stage C
274             k=n-1=4
275 checking v[i]>v[k] or not.
276 ----- v[1]=3, v[4]=1
277 ---++++----- v[1]=3 > v[4]=1
278 decrementing k, k=3
279 ---++++----- v[1]=3 > v[3]=2
280 decrementing k, k=2
281
282 ----- k=2
283 stage D
284 n=5, i=1, k=2
285 ---Always swapped --- before swap, v[i=1]=3, v[k=2]=5
286 ----- swaped v[i=1]=5, v[k=2]=3
287 -----+++++++ The current array is:
288 4 5 3 2 1
289 assigning: k=0
290 checking j < (n+i)/2+1 or not.

```

```
291  n=5, i=1, j=2, k=0
292  -----+++++++ j=2, (n+i)/2+1=4, k=0
293  -----+++++++ n-k-1=4
294  -----+++++++ preparing swap v[j] and v[n-k-1]
295  -----+++++++ before swap v[2]=3, v[4]=1
296  -----+++++++ swaped v[2]=1, v[4]=3
297  -----+++++++ The current array is:
298  4 5 1 2 3
299  incrementing j and k
300  checking j < (n+i)/2+1 or not.
301  n=5, i=1, j=3, k=1
302  -----+++++++ j=3, (n+i)/2+1=4, k=1
303  -----+++++++ n-k-1=3
304  -----+++++++ preparing swap v[j] and v[n-k-1]
305  -----+++++++ before swap v[3]=2, v[3]=2
306  -----+++++++ swaped v[3]=2, v[3]=2
307  -----+++++++ The current array is:
308  4 5 1 2 3
309  incrementing j and k
310  returning 0
311  .....
312  returned to main(). Next permutation is:
313  4 5 1 2 3
314  Press any key to continue . . .
315  =====
316  7
317  =====
318
319
```