

**Source File:** ~/2336/29/lab29.cpp  
**Input:** Under control of `main` function  
**Output:** Under control of `main` function  
**Value:** 2

Given  $n$  distinct elements, generate and list all permutations of those elements. For example, if  $n = 4$  and the digits from 1 through 4 are the elements, the desired output is

1234	2134	3124	4123
1243	2143	3142	4132
1324	2314	3214	4213
1342	2341	3241	4231
1423	2413	3412	4312
1432	2431	3421	4321

There are  $4! = 24$  permutations listed, and in general there are  $n!$  permutations of  $n$  distinct elements.

What is needed is an algorithm that generates one permutation from the immediately preceding permutation without generating any patterns that have to be discarded. For example, how might we operate on the four-digit permutation 1432 to produce 2134, the permutation of next higher (apparent) value? Looking at only this single example is unlikely to suggest an answer, but inspecting the entire sequence of 24 four-digit permutations listed earlier allows us to deduce the following algorithm for transforming a given permutation  $p$ , considered as a one-dimensional array (vector), into the permutation  $p'$ , the next higher permutation:

1. Scan  $p$  from right to left until an element is found that is less than its neighbor to the right. Call the index position of this element  $i$ .
2. Scan again from right to left until an element is found that is greater than the one at  $i$ . Call the position of this element  $j$ .
3. Exchange the  $i^{\text{th}}$  and  $j^{\text{th}}$  elements.
4. Reverse the elements from the  $(i + 1)^{\text{st}}$  position to the right.

For  $n = 4$  and  $p = 1432$ , the first element less than its right neighbor is the 1 in the first position. Then, scanning again from right to left, the first element greater than the 1 in the first position is the 2 in the fourth position. Exchange these elements to get 2431. Now reverse all elements from the second position on, to obtain 2134, which is indeed the successor to 1432 in the earlier listing of all 24 four-digit permutations. For  $n = 7$  and  $p = 3176542$ , the first element less than its right neighbor is the 1 at the second position. Scan again from right to left to find the first element greater than 1, which is the 2 at the seventh position. Exchange the 1 and 2 to obtain 3276541. Now reverse all elements to the right of the second position to obtain  $p' = 3214567$ .

If the  $n$  elements being permuted are distinct, there are  $n!$  permutations; however, if the  $n$  elements are not distinct, there are fewer permutations. For example, suppose the characters `abba` are to be permuted. The above algorithm can be used and shows the permutations to be:

aabb	baab
abab	baba
abba	bbaa

The header file shown in Figure 1 is a specification for the two function templates to be constructed for this assignment. The `nextPermutation` function template implements the algorithm described above. It transforms the input parameter into the next permutation and returns true. If the algorithm fails in the first step, the function reverses the order of the elements and returns false. The other function template is an overloaded output operator for displaying the elements of a vector.

A sample `main` function for testing your functions is shown in Figure 2. Commands to compile, link, and run this assignment are shown in Figure 3. To use the `Makefile` as distributed in class, add a target of `lab29main` to the definition of `targets1srcfile` in your `Makefile`.

```
1  #ifndef LAB29_H
2  #define LAB29_H
3
4  #include <iostream>
5  #include <vector>
6
7  using namespace std;
8
9  template<typename T>
10 bool nextPermutation(vector<T>& v);
11
12 template<typename T>
13 ostream& operator<<(ostream& os, const vector<T>& v);
14
15 #include "lab29.cpp"
16
17 #endif
```

Figure 1. /usr/local/2336/include/lab29.h

```
1  #include <cstdlib>
2  #include <iomanip>
3  #include <string>
4  #include <algorithm>
5  #include <lab29.h>
6
7  using namespace std;
8
9  template<typename T>
10 void generatePermutations(const vector<T>& v, ostream& out);
11
12 int main()
13 {
14     vector<int> a;
15     a.push_back(2);
16     a.push_back(3);
17     a.push_back(0);
18     a.push_back(5);
19     generatePermutations(a, cout);
20
21     vector<double> b;
22     b.push_back(2.3);
23     b.push_back(0.5);
24     cout << fixed << setprecision(1) << showpoint;
25     generatePermutations(b, cout);
26 }
```

Figure 2. /usr/local/2336/src/lab29main.C (Part 1 of 2)

```

27     string inputLine;
28     vector<char> perm;
29
30     while(getline(cin, inputLine))
31     {
32         perm.clear();
33         perm.insert(perm.begin(), inputLine.begin(), inputLine.end());
34         generatePermutations(perm, cout);
35     }
36
37     return EXIT_SUCCESS;
38 }
39
40 template<typename T>
41 void generatePermutations(const vector<T>& v, ostream& out)
42 {
43     vector<T> perm(v), copy;
44     sort(perm.begin(), perm.end());
45     copy = perm;
46
47     out << perm << endl;
48     while (nextPermutation(perm))
49         out << perm << endl;
50     out << endl;
51
52     if (perm != copy)
53         out << "Gotcha; this shouldn't print" << endl;
54 }

```

**Figure 2.** /usr/local/2336/src/lab29main.C (Part 2 of 2)

```

1  newuser@csunix ~> cd 2336
2  newuser@csunix ~/2336> ./getlab.ksh 29
3      * Checking to see if a folder exists for Lab 29. . .No
4      * Creating a folder for Lab 29
5      * Checking to see if Lab 29 has sample input and output files. . .Yes
6      * Copying input and output files for Lab 29
7      from folder /usr/local/2336/data/29 to folder ./29
8      * Checking to see if /usr/local/2336/src/lab29main.C exists. . .Yes
9      * Copying file /usr/local/2336/src/lab29main.C to folder ./29
10     * Checking to see if /usr/local/2336/include/lab29.h exists. . .Yes
11     * Copying file /usr/local/2336/include/lab29.h to folder ./29
12     * Copying file /usr/local/2336/src/Makefile to folder ./29
13     * Adding a target of lab29main to targets1srcfile
14     * Touching file ./29/lab29.cpp
15     * Edit file ./29/lab29.cpp in Notepad++
16  newuser@csunix ~/2336> cd 29
17  newuser@csunix ~/2336/29> ls
18  01.dat      01.out      Makefile     lab29.cpp    lab29.h      lab29main.C

```

**Figure 3.** Commands to Compile, Link, & Run Lab 29 (Part 1 of 2)

<pre> 19  newuser@csunix ~/2336/29&gt; make lab29main 20  g++ -g -Wall -std=c++11 -c lab29main.C -I/usr/local/2336/include -I. 21  g++ -o lab29main lab29main.o -L/usr/local/2336/lib -lm -lbits 22  newuser@csunix ~/2336/29&gt; cat 01.dat 23  acb 24  x 25  qp 26  1234 27  abba 28  loot 29  mm 30  onion 31  error 32  banana 33  newuser@csunix ~/2336/29&gt; cat 01.dat   ./lab29main </pre>					
<pre> 34  0235 35  0253 36  0325 37  0352 38  0523 39  0532 40  2035 41  2053 42  2305 43  2350 44  2503 45  2530 46  3025 47  3052 48  3205 49  3250 50  3502 51  3520 52  5023 53  5032 54  5203 55  5230 56  5302 57  5320 58 59  0.52.3 60  2.30.5 61 62  abc 63  acb 64  bac 65  bca 66  cab 67  cba </pre>	<pre> 68 69  x 70 71  pq 72  qp 73 74  1234 75  1243 76  1324 77  1342 78  1423 79  1432 80  2134 81  2143 82  2314 83  2341 84  2413 85  2431 86  3124 87  3142 88  3214 89  3241 90  3412 91  3421 92  4123 93  4132 94  4213 95  4231 96  4312 97  4321 98 99  aabb 100 abab 101 abba </pre>	<pre> 102 baab 103 baba 104 bbba 105 106 loot 107 loto 108 ltoo 109 olot 110 olto 111 oolt 112 ootl 113 otlo 114 otol 115 tloo 116 tolo 117 tool 118 119 mm 120 121 innoo 122 inono 123 inoon 124 ionno 125 ionon 126 ioonn 127 ninoo 128 niono 129 nioon 130 nnioo 131 nnoio 132 nnooi 133 noino 134 noion 135 nonio </pre>	<pre> 136 nonoi 137 nooin 138 nooni 139 oinno 140 oinon 141 oionn 142 onino 143 onion 144 onnio 145 onnoi 146 onoin 147 ononi 148 ooinn 149 oonin 150 oonni 151 152 eorrr 153 erorr 154 error 155 erroo 156 oerrr 157 orerr 158 orrer 159 orrre 160 reorr 161 reror 162 rerro 163 roerr 164 rorer 165 rorre 166 rreor 167 rrero 168 rroer 169 rrore </pre>	<pre> 170 rrreo 171 rrroe 172 173 aaabnn 174 aaanbn 175 aaannb 176 aabann 177 aabnan 178 aabnna 179 aanabn 180 aananb 181 aanban 182 aanbna 183 aannab 184 aannba 185 abaann 186 abanan 187 abanna 188 abnaan 189 abnana 190 abnnaa 191 anaabn 192 anaanb 193 anaban 194 anabna 195 ananab 196 ananba 197 anbaan 198 anbana 199 anbnaa 200 annaab 201 annaba 202 annbaa 203 baaann </pre>	<pre> 204 baanan 205 baanna 206 banaan 207 banana 208 bannaa 209 bnaaan 210 bnaana 211 bnanaa 212 bnnaaa 213 naaabn 214 naaanb 215 naaban 216 naabna 217 naanab 218 naanba 219 nabaan 220 nabana 221 nabnaa 222 nanaab 223 nanaba 224 nanbaa 225 nbaaan 226 nbaana 227 nbanaa 228 nbnaaa 229 nnaaab 230 nnaaba 231 nnabaa 232 nnbaaa 233 </pre>
<pre> 234 newuser@csunix ~/2336/29&gt; cat 01.dat   ./lab29main &gt; my.out 235 newuser@csunix ~/2336/29&gt; diff 01.out my.out 236 newuser@csunix ~/2336/29&gt; </pre>					

Figure 3. Commands to Compile, Link, &amp; Run Lab 29 (Part 2 of 2)