Program 2 – Generating Permutations

CSC2400

Assigned: 10/10/2018

Due: 10/26/2018 (by 11:59 PM)

DESCRIPTION

Write a C++ program implemented in a single source file named **permutations.cpp** that uses **two** decrease-and-conquer algorithms for generating permutations. The two algorithms are Johnson Trotter and Lexicographic Permute. The pseudocode for these algorithms are provided in this assignment. Each permutation should be printed to the screen in the specific way as the example output at the bottom of this assignment.

JOHNSON TROTTER

```
1:  ALGORITHM JohnsonTrotter(n)
2:      // Johnson-Trotter Algorithm for generating permutations
3:      // Input: A positive integer n
4:      // Output: List of all permutations of {1, ..., n}
5:      P[1, ..., n] ← (←1, ←2, ..., ←n)
6:      while P contains a mobile element do
7:          Find the largest mobile element k
8:          swap k with its adjacent element that k's arrow points to
9:          reverse the arrow of all elements larger than k
10:         add P to the list of permutations
11:     end while
12: end ALGORITHM
```

LEXICOGRAPHIC PERMUTE

```
1:  ALGORITHM LexicographicPermute(n)
2:      // A Lexicographic algorithm for generating permutations
3:      // Input: A positive integer n
4:      // Output: List of all permutations of {1, ..., n} in lex order
5:      while P contains two consecutive elements in increasing order do
6:          let i be the largest index such that P[i] < P[i + 1]
7:          let j > i be the largest index such that P[i] < P[j]
8:          swap P[i] with P[j]
9:          reverse P[i + 1, ..., n]
10:         add P to the list of permutations
11:     end while
12: end ALGORITHM
```

NOTE: so element in i needs to be GREATER THAN element in i doesn't have to contain the LARGEST value to the right of i. Also, j begins its search at end of array & moves to left until reaches i+1

MODIFICATIONS TO THE ALGORITHMS

Where the algorithm says the input is a **positive integer n**, you are to write the code where **n** is an array of **1 to n** characters. The characters can contain character digits '0' through '9' or characters 'A' through 'Z'.

You may assume that the user has entered in either digits 0 through 9 or uppercase characters A through Z (you do not have to do error checking with user input).

You may also assume that the input entered by the user is a sorted array in increasing order.

So, your program should be able to generate permutations of inputs such as "1234", "ACT", "ABCDE", "2468", etc.

DON'T CHEAT & GET HELP

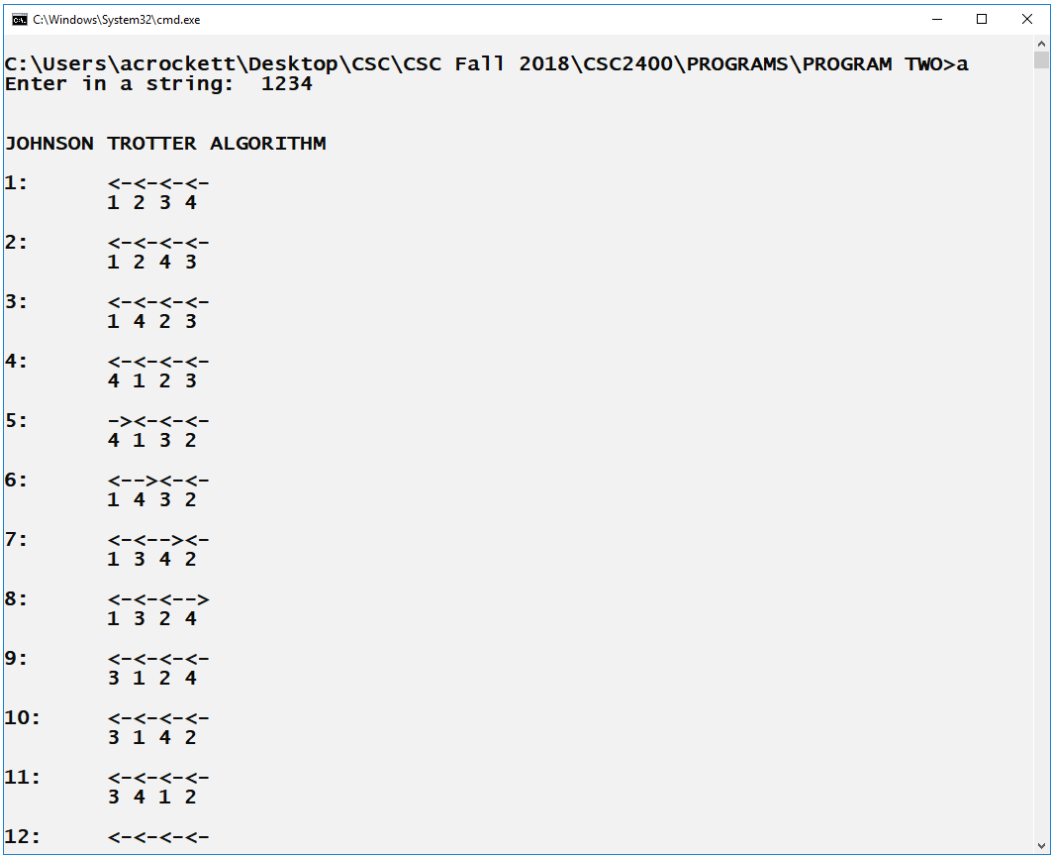
Make sure you review the "Get Help & Don't Cheat!" section of our class syllabus so that you are not charged with academic misconduct. I am aware that many implementations of these algorithms are available online. However taking solutions off of websites and forums online and submitting them as your own work is considered cheating in CSC2400. The assignment is to be able to create your own implementation of these algorithms, not simply just to find an implementation online.

WHAT TO TURN IN

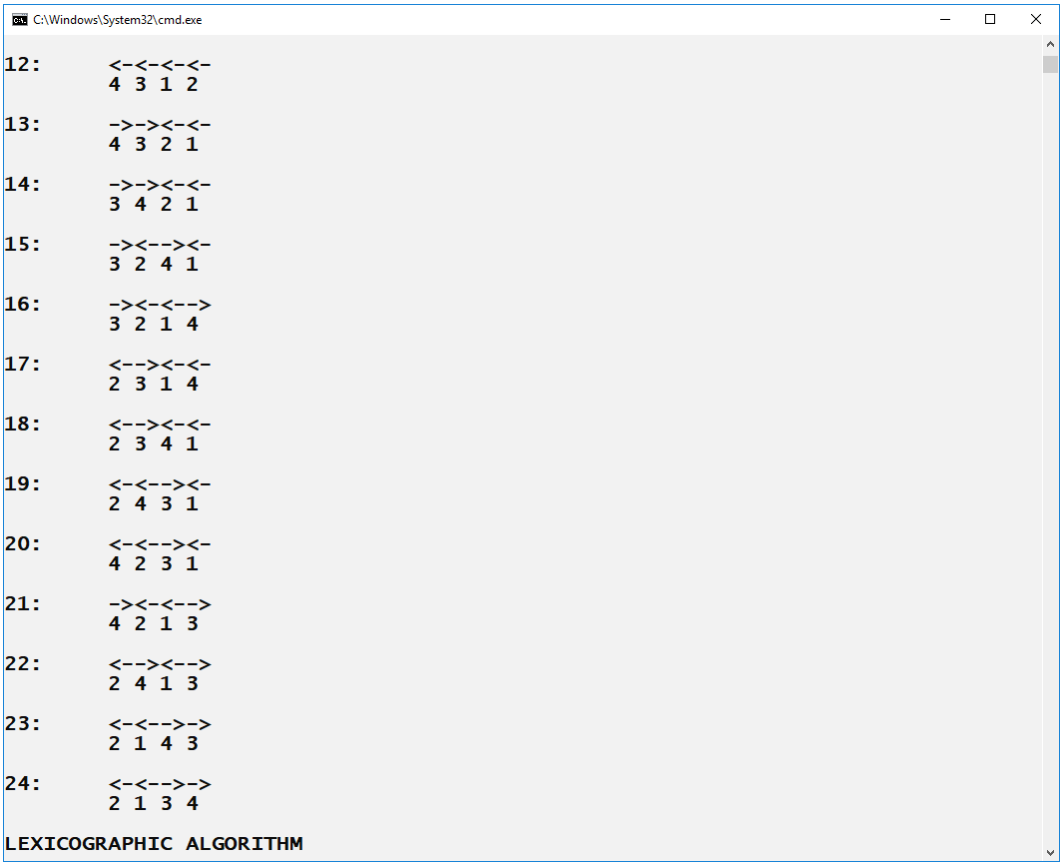
Zip your **permutations.cpp** file and then upload it to the program 2 assignment folder in ilearn by the due date.

SAMPLE OUTPUT

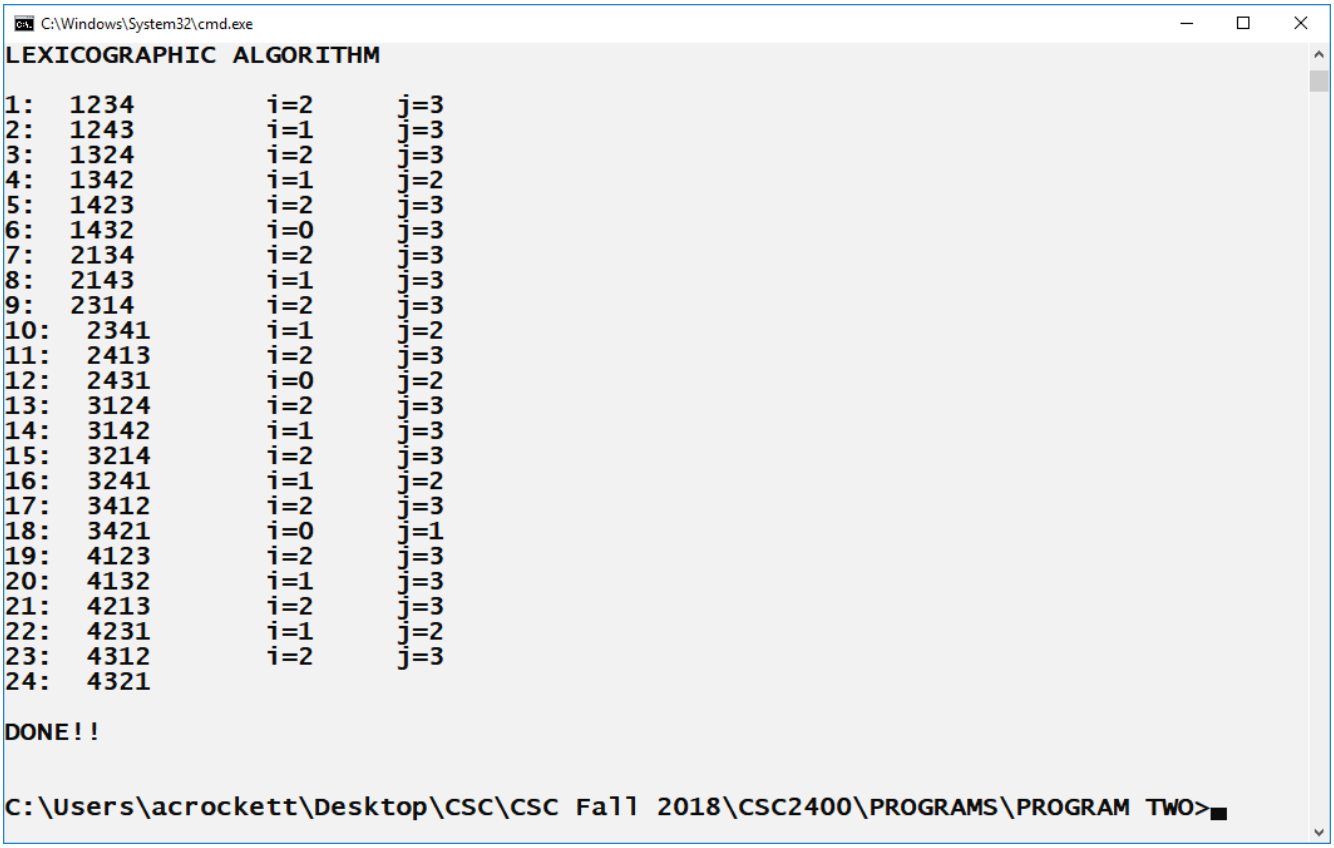
EXAMPLE ONE



```
C:\Windows\System32\cmd.exe

C:\Users\acrockett\Desktop\CSC\CSC Fall 2018\CSC2400\PROGRAMS\PROGRAM TWO>a
Enter in a string:  1234


JOHNSON TROTTER ALGORITHM

1:      <-<-<-<-
        1 2 3 4

2:      <-<-<-<-
        1 2 4 3

3:      <-<-<-<-
        1 4 2 3

4:      <-<-<-<-
        4 1 2 3

5:      ->-<-<-<-
        4 1 3 2

6:      <-->-<-<-
        1 4 3 2

7:      <-<-->-<-
        1 3 4 2

8:      <-<-<-->
        1 3 2 4

9:      <-<-<-<-
        3 1 2 4

10:     <-<-<-<-
        3 1 4 2

11:     <-<-<-<-
        3 4 1 2

12:     <-<-<-<-
```



```
C:\Windows\System32\cmd.exe

12:     <-<-<-<-
        4 3 1 2

13:     ->->-<-<-
        4 3 2 1

14:     ->->-<-<-
        3 4 2 1

15:     ->-<-->-<-
        3 2 4 1

16:     ->-<-<-->
        3 2 1 4

17:     <-->-<-<-
        2 3 1 4

18:     <-->-<-<-
        2 3 4 1

19:     <-<-->-<-
        2 4 3 1

20:     <-<-->-<-
        4 2 3 1

21:     ->-<-<-->
        4 2 1 3

22:     <-->-<-->
        2 4 1 3

23:     <-<-->->
        2 1 4 3

24:     <-<-->->
        2 1 3 4

LEXICOGRAPHIC ALGORITHM
```



Note: I do want you to print out the arrow directions for the Johnson Trotter algorithm as shown in the output above. Also, I do want the value for i & j to be3 printed out for the lexicographic algorithm.

EXAMPLE TWO