Hamza Shaikh 40129291
Alex Kofman 40131871

Pseudo code version 2

//Algorithm: **permute(String shortStr, String longStr)**
//Input: Two Strings shortStr and longStr
Assume **findCelling(temp, index)** that takes an array of characters temp and an integer index and **swap(str,i,j)** that takes an array of characters and two integers i and j respectively

```
//Output
Set shortStr to array of char
Temp <- shortStr
Sort temp in ascending order
foundIndex <- index of shortStr in longStr
If foundIndex != -1 Then
        print the match and corresponding index
//declare variables
index<-0
lowest <- 0
While true
        For i<- 0 to i<temp length do
                If temp[i] < temp[i+1] then
                    lowest <- i
                End
        End
Index <- lowest
J <- findCeiling(temp,index)
If (j == index) then
        Break
swap(temp, index,j)
a <- string.valueOf(temp)
//sort the substring
b<- a.substring(index+1).toCharArray()
Sort b in ascending order
b<- a.substring(index+1) +String.valueOf(b)
temp <- a.toCharArray
foundIndex <- index of shortStr in longStr
If foundIndex != -1 Then
        print the match and corresponding index
Print out the content of temp
End
```

//Explanation: The time complexity of this algorithm is of O(n^2) as we look at the worst case scenario in which we have a while loop that keeps iterating as long as all permutations are printed (n times) and this while loop contains a for loop, a call to the findCeiling and swap function which each have a time complexity of n by neglecting constants. Therefore, if we add them we obtain a time complexity of O(n*n)==> O(n^2).

```
//Algorithm: findCeilling(temp,index)
//Input:array of characters temp and an integer index
//Output
k<- index
test<-temp[index]
While k< temp.length -1 do
        If temp[index]<temp[k+1] then
                Index <- k+1
                break
        end
        k++
end
k<-index
While k<temp.length-1 do
        If temp[index]>temp[k+1] and temp[k+1]>test then
                index<-k+1
        End
        k++
End
Return index

//Algorithm: swap(str,i,j)
//Input: array of characters and two integers i and j respectively
//Output:
temp<- str[i]
str[i]=str[j]
str[j]=temp
```