Problem Set 10

1. Given n distinct elements from a totally ordered set, we may number the $n!$ permutations of them according to the lexicographic order. For example, for the three integers $5, 7, 9$, we may number their permutations as follows:

$$0 : (5, 7, 9)$$
$$1 : (5, 9, 7)$$
$$2 : (7, 5, 9)$$
$$3 : (7, 9, 5)$$
$$4 : (9, 5, 7)$$
$$5 : (9, 7, 5)$$

   This numbering, a bijection from permutations to non-negative integers, is known as the **Cantor Expansion**.

   (a) Write a function that takes by reference a vector of integers that represents a permutation of n distinct integers, manipulates the vector so that it contains the next permutation, "next" understood to be in the ordering we just discussed. For this problem, you are not allowed to use the STL function `next_permutation()`.

   (b) Write a function that takes a vector of integers that represents a permutation of n distinct integers, returns the index of this permutation in the ordering we just discussed. For example, if the input is $(7, 9, 5)$, it should return 3. Your algorithm must run in time polynomial in n. In particular, it must not call the function in part (a) on a sorted vector a number of times until the vector becomes equal to the vector. That algorithm runs in time $\Omega(n!)$, which is not polynomial.

   (c) Write a function that takes by reference a vector of n integers and another integer C, and manipulates the vector so that it contains the permutation whose index is equal to C. Your algorithm must run in time polynomial in n.

2. (a) You are given n points in the 2-dimensional Euclidean space. Suppose an ant starts from the origin $(0, 0)$ and runs to each of these points in a certain order. Calculate the shortest distance the ant has to run. (The ant does not need to come back to the origin.)

   As input, you are given an integer n, followed by n lines, each line containing two reals indiciating the x and y coordinates of the i-th point. As output, you should output the minimum distance the mouse has to run. In the answer you should keep two digits after the decimal point.

   Example input:

```
4
```

```
1 1
1 -1
-1 1
-1 -1
```

Example output:

```
7.41
```

(b) (Bonus) Solve the same problem with a program whose running time is $O(2^n \cdot \text{poly}(n))$, where $\text{poly}(n)$ is a polynomial function of n. Note that a running time of $\Omega(n!)$ does not satisfy this requirement. (See Luogu P1433.)