Problem 1

Write a program that accepts a string of length three as input and print all possible three-letter strings that can be formed using the characters in the original string. Repetition is allowed. Print the strings in alphabetical order, one string on each line. You can assume that all characters in the input string will be unique.

Test Case :

	Input	Expected Output
1.	abc	aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc

Problem 2

This problem is about reversing a square matrix along row or column:

Matrix	Reverse along row	Reverse along column
1,2	3,4	2,1
3,4	1,2	4,3

The first line of the input will be an integer n, which denotes the dimension of the square matrix. Each of the next n lines in the input will have a sequence of n comma-separated integers, you can assume that there will not be any space after comma. The last line in the input will be one of these two words: row or column. If it is “row”, then reverse the matrix along the row, else, reverse it along the column.

Print the reversed matrix as output: each line should contain one row of the matrix as a sequence of comma-separated integers.

Test Case:

1. Input
   2
   1,2
   3,4
   row

   Expected Output
   3,4
   1,2

2. Input
   2
   1,2
   3,4
   column

   Expected Output
   2,1
   4,3

Problem 3

Accept two positive integers start and end as input. Construct the following sets:

1. A is the set of positive integers from start to end (endpoints inclusive) divisible by 3.
2. B is the set of positive integers from start to end (endpoints inclusive) divisible by 5.

Find the set of all integers that are:

- divisible by 3 or 5, and store it in the variable o1
- divisible by both 3 and 5, and store it in the variable o2
- divisible by 3 but not divisible by 5, and store it in the variable o3
- divisible by 5 but not divisible by 3, and store it in the variable o4

Note that each bullet corresponds to a separate set. You have to accept start and end as input and create these four sets. You do not have to print the output to the console.

Test Case:

1. Input
   1
   10

   Expected Output
   [3,5,6,9,10]
   []
   [3,6,9]
   [5,10]

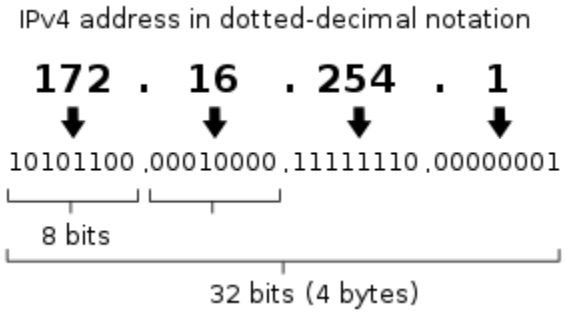
2. Input
   3
   25

   Expected Output
   [3,5,6,9,10,12,15,18,20,21,24,25]
   [15]
   [3,6,9,12,15,18,21,24]
   [5,10,20,25]

Problem 4

An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

For example:



Classification of IPv4

CLASS	From	To
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

Write a program to read all IP addresses (binary format) before termination line END from ***ipaddress.txt*** file and after converting into decimal format , print the output in following format.

Output

A = count of ip that belongs to class A
B = count of ip that belongs to class B
C = count of ip that belongs to class C
D = count of ip that belongs to class D
E = count of ip that belongs to class E

Test Case:

Sample Input file:

11111111.10101010.01010101.11001100
11110000.00000000.11111111.00000001
10001000.01010101.10100000.10101000
11000000.11111111.11111111.11111111
11100000.10101010.00000000.11111111
01000000.10101000.01010101.10000000
END

Output corresponding to the sample input file:

A = 1
B = 1
C = 1
D = 1
E = 2

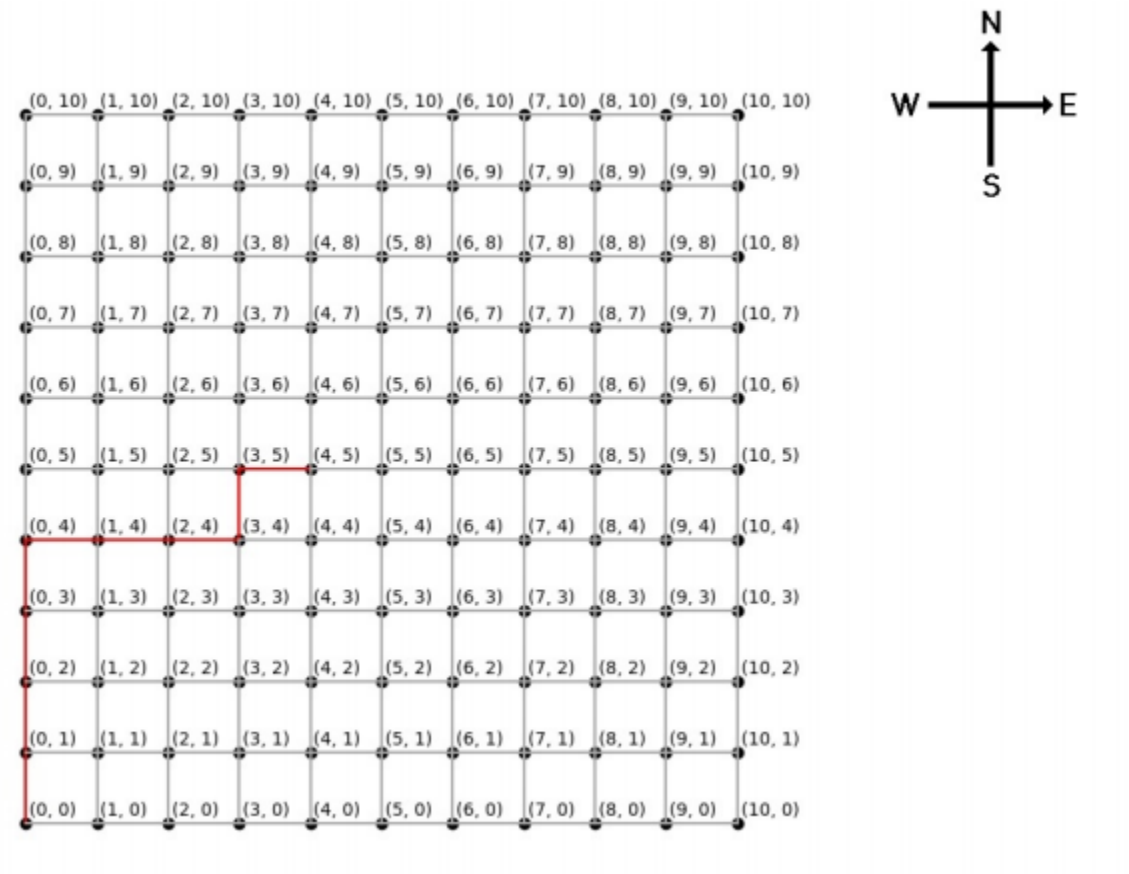
Problem 5

The below image represents a grid having 11x11 nodes numbered from 0 to 10.

- Distance between one node to the next connected node is 1 unit.
- One can go in any direction, each letter counts as 1 unit in each direction.

    - **N** North
    - **S** South
    - **E** East
    - **W** West

The below graph shows the path of "**NNNNEEENE"** starting from (0,0)



Write a program to take a string as input from the user and print the total distance travelled.

Test Case:

1. Input
   NNNNEEENE
   Expected Output
   9

2. Input
   NEWS
   Expected Output
   4