Assignment #1, Advanced Programming Lab. 2016 Spring Semester

Due date: 2016/04/01, 23:59:59

1. Write a program that reads a word and prints each character along with the next two characters on a separate line form the first character to the last one. If the next letter is NULL, then go back to the first letter of the word. The output should be as follows.

Input: apple app ppl ple lea eap	Input: a aaa

2. Write a program that reads the coordinates of the three points of a rectangle and finds a rest of the coordinates. Each side of the rectangle is assumed to be parallel to the x-axis and y-axis. Displays the resulting coordinate. Note that the program should have an exception handling to check whether the input is a rectangle or not.

Input [1]: 3 5 [2]: 5 5 [3]: 5 3 Output [4]: 3 3	Input [1]: 3 5 [2]: 5 5 [3]: 4 3 Output Not rectangle

3. Write a program that reads a string and calculates the frequency of characters. For example, when N_{all} is the total number of characters and N_a is the number of character 'a', the frequency of character 'a' can be calculated as: N_a/N_{all}. Displays the result in the following format.
   Assumption: The input string consists of the lowercase alphabet.

```
Input: apple banana
Output
a      0.363636
b      0.090909
e      0.090909
…
```

4. Write a program that reads a random word from input file and hide two letters at random. A user should correct the hidden letters and the program shows the letter whenever the user correct the answer. A user has four chances for correcting the hidden letters. Use your input file(only one word per line).
   For example,

== Start == r_ver_e >> a r_ver_e >> e rever_e >> s reverse == End ==	== Start == di_ru_t >> a di_ru_t >> t di_ru_t >> o di_ru_t >> s disru_t Fail; disrupt == End ==

5. Write a program that reads an integer number, n and find the smallest number that has the same set of digits as n and is greater than n. If any number cannot be found under the conditions, then print “Not possible”.
Constraints: 1<= n < 10000

Input: 1215 Output: 1251	Input: 14355 Input: 543 Not possible

6. Given an unsorted array of positive integers, find the number of triangles that can be formed with three different array elements as lengths of three sides of triangles. You should be use dynamic allocation for the array. Note that the sum of two side lengths of a triangle is always greater than the third side.

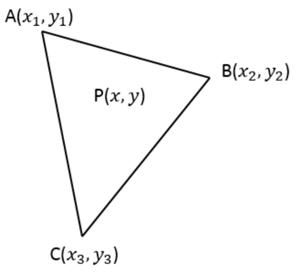
```
Input the size of array: 4
Input: 5 7 2 4
Output: 2
```

7. Input an integer number in the range of [1,10000], and return the answer of following questions.
   a. Can it be divided by 7, 11, or 13?
   b. Is the sum of every digit odd or even?
   c. Is it a **prime number**?

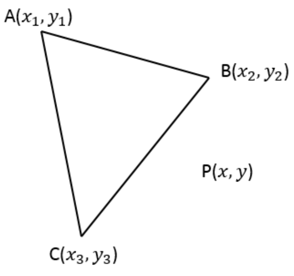
Note: A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. For example,

```
The number is 2282
It can be divided by 7, the result is 326
It cannot be divided by 11
It cannot be divided by 13
The sum of every digit is 14, it is even number
It is not a prime number
```

8. Write a program to detect whether a point is in the area of a specified triangle. The user is firstly asked to enter 3-point coordinates which can form a triangle, and then enter another point coordinate, the program should make the judgement whether this point is in the area of the triangle surrounded by the three point coordinates.
(Note: If the point is in any side of the triangle, it should be regarded that the point is in the triangle)



The Point is in the triangle



The Point is not in the triangle

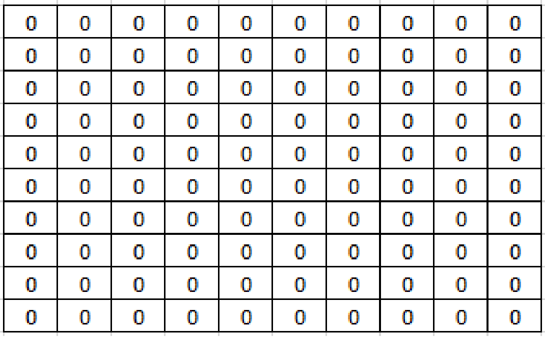
9. Write a program to print the calendar into a file named **Calendar.txt**, the user is asked to specify the year and the month. The year should be limited in the range of [2010, 2020]. Note: The first day of 2010 is Friday.
For example,

```
Enter the year: 2015
Enter the month: 7

                1     2     3     4     5
 6     7     8     9    10    11    12
13    14    15    16    17    18    19
20    21    22    23    24    25    26
27    28    29    30    31
```

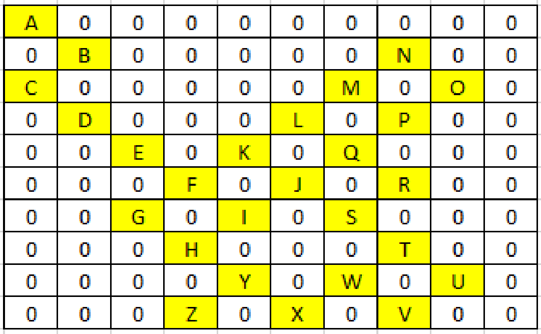
10. Write a program that produces a 10x10 2D character array (initialized as char ‘0’). Starting from letter ‘A’ at the top left of 2D array, and move to another one-step position randomly in four (top left, top right, left bottom, and right bottom) diagonal directions. Note that any character cannot be placed outside of 2D array plane and two characters cannot be assigned to the same position. In case that there is no way to go, other four (up, bottom, left, right) directions for once can be selected and return to the rules before. The program stops when it runs out of 26 letters (‘A’ to ‘Z’). At last, the program should print out the array.
(Note: For the condition that there is no way to go in both methods, throw the exception “The Program Stops!”)
For example,

1. Initialization



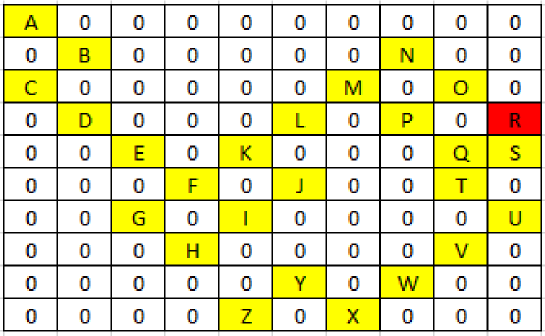
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2. Regular example



A	0	0	0	0	0	0	0	0	0
0	B	0	0	0	0	0	N	0	0
C	0	0	0	0	0	M	0	O	0
0	D	0	0	0	L	0	P	0	0
0	0	E	0	K	0	Q	0	0	0
0	0	0	F	0	J	0	R	0	0
0	0	G	0	I	0	S	0	0	0
0	0	0	H	0	0	0	T	0	0
0	0	0	0	Y	0	W	0	U	0
0	0	0	Z	0	X	0	V	0	0

3. Exception example (The red ‘R’ means there is no “way” to go, so it chooses randomly one of up, bottom, left, and right)



A	0	0	0	0	0	0	0	0	0
0	B	0	0	0	0	0	N	0	0
C	0	0	0	0	0	M	0	O	0
0	D	0	0	0	L	0	P	0	R
0	0	E	0	K	0	0	0	Q	S
0	0	0	F	0	J	0	0	T	0
0	0	G	0	I	0	0	0	0	U
0	0	0	H	0	0	0	0	V	0
0	0	0	0	0	Y	0	W	0	0
0	0	0	0	Z	0	X	0	0	0

11. Write a program which produces a "Number Square", The user is asked to enter the length of square side which is an integer number in the range of [3, 15]. Starting from the up left of the square, to the center of square, the numbers are listed incrementally, circumferentially in clockwise order.
    For example,

```
Please enter the length of square side: 3
1   2   3
8   9   4
7   6   5
Please enter the length of square side: 4
 1    2    3    4
12   13   14    5
11   16   15    6
10    9    8    7
Please enter the length of square side: 5
 1    2    3    4    5
16   17   18   19    6
15   24   25   20    7
14   23   22   21    8
13   12   11   10    9
```