

ชื่อ ..... รหัสประจำตัว ..... หมู่ ..... / ..... เลขที่ ..... วันที่ .....

## ใบงานที่ 10

### ข้อตกลงการส่งคำตอบ

- Lab sheet อยู่ในรูปแบบไฟล์อิเล็กทรอนิกส์ ทั้งหมด
- การส่งงาน
  - ห้ามนำข้อความใน Lab Sheet และ Lecture เผยแพร่ออกไปสู่สาธารณะ
  - นิสิตจะต้องใส่ส่วนของ comment เพื่อแสดงรายละเอียด ชื่อไฟล์ ผู้เขียน รหัสนิสิต หมู่เรียน ลำดับที่ วันที่ **ทุกไฟล์ (ทั้ง ไฟล์โค้ด และ ไฟล์คำตอบบรรยาย)**
  - สำหรับโจทย์ ที่มี (filename.cs) หลังชื่อข้อ ให้บันทึกไฟล์โค้ดโปรแกรมตามชื่อที่กำหนด
  - สำหรับโจทย์ ที่มีคำตอบบรรยาย ให้นิสิตรวมคำตอบ (ทุกข้อย่อย) เพียงไฟล์เดียว ตั้งชื่อตามที่โจทย์กำหนด

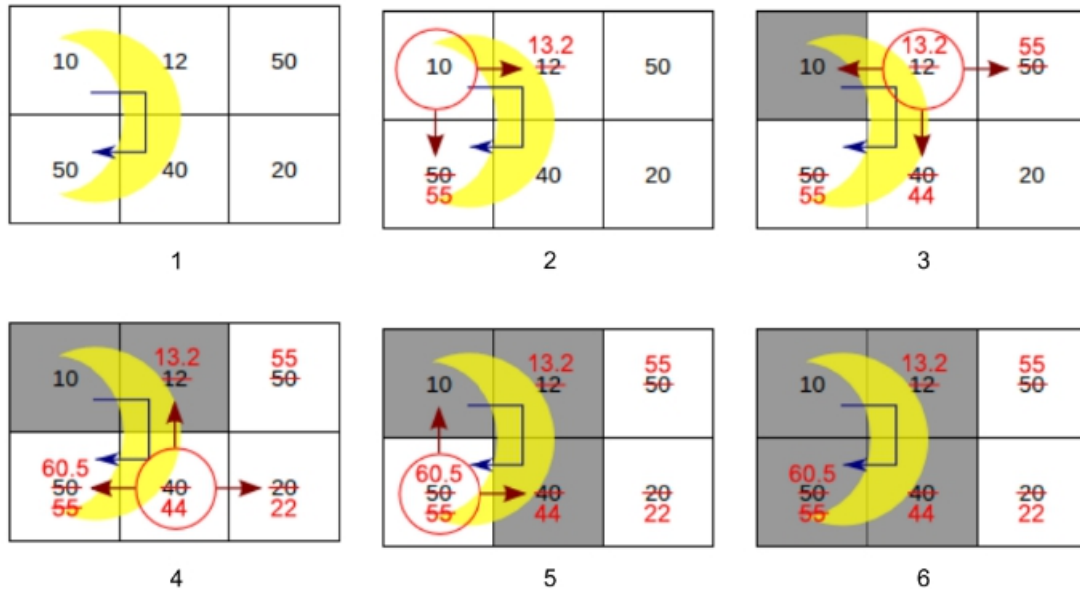
```
// ===== Edit Student Detail Here =====  
// Author: Tom Jerry  
// Id: 5900000001  
//Program: XXX.cs  
// Date: 11 November 2016  
//=====
```

ข้อ 1.1 )  
    วิธีการแก้ไข error โดยการเติม ; ในบรรทัดที่ 20

ข้อ 1.2 )  
    วิธีการแก้ไข error โดยการเติม ; ในบรรทัดที่ 21

1. [huangjui.cs] กาลครั้งหนึ่งซึ่งไม่นานมานัก มีเศรษฐีจอมงกท่านหนึ่ง หลังจากร่ำรวยจากการทำธุรกิจเกมสออนไลน์แล้ว เกิดอยากจะเสริมสร้างดวงชะตาชีวิตให้แข็งแกร่งขึ้นไปอีก จึงไปหาซินแสจีนท่านหนึ่งเพื่อขอคำปรึกษาแนะนำ ซินแสท่านนั้นได้แนะนำให้เศรษฐีจอมงกไปหาซื้อที่ดินจำนวน 4 แปลง เพื่อสร้างเป็นสวนจันทระปลูกเสริมดวงชะตา โดยการซื้อที่ดินนั้น จะต้องซื้อที่ดินที่แปลง และติดกันจำนวน 4 แปลงเป็นรูปพระจันทร์เสี้ยวหันท้าย (ตามสัญลักษณ์บนหน้าผากของท่านเปา ซึ่งเกิดปีนักษัตรเดียวกันกับเศรษฐีจอมงก) และจะต้องซื้อเป็นลำดับต่อเนื่อง "วนตาม" หรือ "วนทวน" เข็มนาฬิกาเท่านั้น แต่หลังจากผู้ขายที่ดินได้ทราบว่าเศรษฐีจอมงกจะมาซื้อที่ดิน จึงตกลงร่วมกันว่า ถ้าเศรษฐีจอมงกซื้อที่ดินแปลงใดไปก็ตาม ที่ดินแปลงที่ยังว่างอยู่แปลงที่ติดกันทั้ง 4 ทิศ (เหนือ ใต้ ออก ตก) จะถูกตั้งราคาขายขึ้นจากราคาเดิมอีก 10% เสมอ

ให้นิสิตพัฒนาโปรแกรมเพื่อช่วยให้เศรษฐีจอมงกซื้อที่ดินได้ 4 แปลงติดกันต่อเนื่องเป็นรูปพระจันทร์เสี้ยวตามที่ซินแสจีนให้คำแนะนำไว้ ได้ในราคาถูกที่สุด



\*\*จากรูปด้านบนเป็นตัวอย่างการคำนวณค่าใช้จ่ายเมื่อเลือกซื้อที่ดินตามคำแนะนำของซินแส จากที่ดิน 6 แปลงที่อยู่ติดกันแบบอาร์เรย์ 2x3 ซึ่งหากซื้อในลักษณะนี้ จะมีค่าใช้จ่าย  $10 + 13.2 + 44 + 60.5 = 127.7$  (แต่หากเป็นบริเวณเดียวกัน แต่ซื้อที่ดินตามลำดับทวนเข็มนาฬิกาจะมีค่าใช้จ่าย  $50 + 44 + 13.2 + 12.1 = 119.3$ ) \*\*

### Input

บรรทัดที่หนึ่ง และ สอง รับค่า m (แถว) และ n (หลัก) ตามลำดับ (โดยที่  $m \geq 2, n \geq 2$ )

บรรทัดที่สาม ถึงบรรทัด m+2 รับค่าราคาที่ดินของแต่ละแถว โดยที่แต่ละบรรทัดจะประกอบด้วยที่ดิน n หลัก ซึ่งแต่ละหลักคั่นด้วยเว้นวรรค

### Output

บรรทัดเดียว ราคาต่ำสุดที่เศรษฐีจอมงกจะต้องจ่ายเงินเพื่อซื้อที่ดินให้ได้ 4 แปลง ทำสวนจันทระตามที่ท่านซินแสแนะนำ

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:

Output:

2	119.3
3	
10 12 50	
50 40 20	

2. [specialCouple.cs] ณ โรงเรียนมัธยมยอดหญิงแห่งหนึ่ง นักเรียนมีความเชื่อแปลกๆว่าเพื่อนที่เข้ากันได้ดีนั้น ต้องมีเลขประจำตัวที่มีลักษณะพิเศษด้วยซึ่งเรียกกันว่า **คู่พิเศษ** การเป็นคู่พิเศษเชื่อว่าจะทำให้ทั้งคู่สนิทกันได้ยาวนานกว่าเพื่อนคนอื่นๆ หรืออาจมีความสัมพันธ์แบบที่เรียกว่า **"ไม่ต้องรู้ว่าเราคบกันแบบไหน ;P"** ซึ่งในโรงเรียนแห่งนี้มีเลขประจำตัวเพียง 4 หลักเท่านั้น การเป็นคู่พิเศษได้จะต้องมีเลขประจำตัวอย่างน้อย 2 ใน 4 หลักที่ซ้ำกัน ตัวอย่างของคู่พิเศษเช่น 1420 กับ 1580 , 5210 กับ 9218 , 3482 กับ 7442 เป็นต้น ให้นิสิตพัฒนาโปรแกรมบอกจำนวนคู่พิเศษจากเลขประจำตัวที่ผู้ใช้ป้อนเข้าสู่โปรแกรม

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:	Output:
How many students : 5 2357 2367 1367 0041 1741	4 special couples
How many students : 6 4109 0156 3166 4507 4582 0506	5 special couples
How many students : 5 3456 9843 2345 3756 2341	2 special couples

3. [wordSearch03.cs] ให้นิสิตพัฒนาโปรแกรมค้นหาคำจากประโยคที่รับมา โดยคำที่พบนั้นสามารถมีจุดที่ต่างกันได้อย่างมากที่สุดเพียง 1 ตำแหน่ง (**Case insensitive**) แต่ถ้าพบคำที่เหมือนกันทุกตัวอักษรอยู่ด้วย ก็จะเลือกแสดงเฉพาะคำที่ถูกต้อง (ตัวอักษรเรียงเหมือนกันเป๊ะๆ) เท่านั้น

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:	Output:
Input Sentence: How are you today Input Word: TaDay	Found How are you [t?day]
Input Sentence: Hello november Input Word: December	Not found
Input Sentence: fun and fat Input Word: FAn	Found [F?n][?an]d [fa?]
Input Sentence: get git GAT Input Word: GOT	Found [g?t] [g?t] [G?T]
Input Sentence: get git GAT GET Input Word: GET	Found [get] git GAT [GET]

4. [fibonacci.cs] ให้นิสิตพัฒนาโปรแกรมเพื่อแสดงค่า fibonacci พจน์ที่ n ( $a_n$ ) โดยใช้ **recursive function** โดยที่ลำดับ fibonacci มีลำดับดังนี้ 0 1 1 2 3 5 8 13 21 34 55... และ **comment อธิบายใต้โปรแกรม**

1. การเขียนแบบ recursive function มีความแตกต่างจากการเขียนแบบลูปอย่างไร เพราะเหตุใด
2. หากโปรแกรมทำ recursion ไปเรื่อยๆ โดยไม่หยุด จะเกิดอะไรขึ้น (**หาคำตอบจากอินเทอร์เน็ต**)

Hint กำหนดให้  $a_0 = 0$ ,  $a_1 = 1$  และ  $a_n = a_{n-1} + a_{n-2}$

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:	Output:
n: 6	a6 = 8
n: 10	a10 = 55
n: 15	a15 = 610

## 5. [countIdentifier.cs] ให้หีสพัฒนาโปรแกรมเพื่อนับจำนวน identifier ในโค้ดภาษา C#

HINT: ลองนำความรู้เกี่ยวกับ data collection มาประยุกต์ใช้

รายชื่อคำที่เป็น Keyword ในภาษา C# <https://msdn.microsoft.com/en-us/library/x53a06bb.aspx>

สามารถศึกษาวิธีการอ่านไฟล์จากโค้ดด้านล่าง หรือ ใน Lecture Array in C#

```
using System;
using System.IO;
class count {
    static void Main(string[] args) {
        StreamReader SR = File.OpenText(args[0]);
        string s = SR.ReadLine();
        while(s != null) {
            //insert your code here
            //...
            Console.WriteLine(s);
            s = SR.ReadLine();
        }
        SR.Close();
    }
}
```

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:

Output:

\$ mono countIdentifier.exe example01.cs	[System=1] [insertion=1] [Main=1] [Console=10] [Write=4] [input=2] [ReadLine=1] [inputArr=4] [Split=1] [inputs=16] [Length=4] [i=9] [Parse=1] [insertionSort=2] [printArr=3] [run=6] [ReadKey=2] [WriteLine=3] [Convert=1] [ToString=1] [now=5] [sort=7] [temp=2]
\$ mono countIdentify.exe example02.cs	[System=3] [cpe3=1] [Main=1] [Console=5] [Write=2] [n=36] [Parse=1] [ReadLine=1] [cpe=39] [mkPattern=2] [printGif=2] [row=56] [drawLeftArrow=2] [drawC=2] [drawP=2] [drawE=2] [drawRightArrow=2] [count=12] [countUp=10] [Clear=1] [Length=1] [nbSpace=21] [WriteLine=1] [updateCount=2] [updateCountStatus=2] [Threading=1] [Thread=1] [Sleep=1] [TimeSpan=1] [FromSeconds=1] [TotalMilliseconds=1] [maxCount=2] [nbX=21]
\$ mono countIdentify.exe example03.cs	[System=1] [max=9] [Main=1] [Console=9] [Write=7] [rawInput=2] [ReadLine=1] [inputArr=4] [Split=1] [inputs=13] [Length=9] [i=20] [Parse=1] [freq=13] [mkFreq=2] [printResult=2] [findMax=2] [j=5] [WriteLine=1] [Math=1] [Max=1] [now=9] [countFreq=7] [next=5] [printArr=1] [arr=3]

สามารถโหลดไฟล์ตัวอย่างที่ใช้ทดสอบได้ที่ /tmp/example0X.cs

6. [stringDetection.cs] ให้นิสิตพัฒนาโปรแกรมเพื่อแสดงผลว่า string ที่รับ input เข้ามาผิดหรือไม่ โดยมีเงื่อนไขดังนี้

- 1) string ที่รับเข้ามามีรูปแบบ wRxLyMz เช่น abcdRdcbaLacdMdca
- 2) w,y เป็น string ที่ประกอบด้วยตัวอักษร a,b,c หรือ d ความยาวเท่าไรก็ได้
- 3) x,z เป็น string ที่เป็นส่วนกลับของ string w และ y ตามลำดับ
- 4) กำหนดให้ผู้ใช้จะใส่ string เป็นตัวอักษรภาษาอังกฤษพิมพ์เล็กทั้งหมด ยกเว้นตัวอักษรพิมพ์ใหญ่ 3 ตัวคือ R L M ซึ่งจะใส่มาแค่อย่างละตัวอักษรเท่านั้น
- 5) ให้ใช้โครงสร้างของ Stack หรือ Queue (สามารถศึกษาได้จาก Lecture Collection & Recursion in C#)

ตัวอย่าง string ที่ถูกต้องตามเงื่อนไขทุกข้อข้างต้น

aaabbbRbbbbaaaLcdcdMdcdc      aaaRaaaLcbcbcbMbcbbcbcb  
adbcaRacbdaLadabMbada      acadabRbadacaLM

ตัวอย่าง string ที่ผิดจากเงื่อนไขข้างต้น

rabadRdabarLttttMtttt (w,y ผิด)      aaaRaaaLtreeMthree (y,z ผิด)  
abcdRabcdLabcdMabcd (x,z ผิด)      bbbRdddLaaaMaaa (x ผิด)

HINT: จำนวน item ที่อยู่ใน queue หรือ stack สามารถหาได้โดยใช้ queueName.Count หรือ stackName.Count

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:

Output:

Input : badRdabLcadMdacc	w string is correct format. x string is correct format. y string is correct format. z string is not correct format.
Input : abcdRdcbaLrmdMdrn	w string is correct format. x string is correct format. y string is not correct format. z string is not correct format.
Input : abcRcbaLdcbMbcd	w string is correct format. x string is correct format. y string is correct format. z string is correct format.
Input : rlmdRdmlrLoverMlower	w string is not correct format. x string is correct format. y string is not correct format. z string is not correct format.

## 7. (โบนัส) [crossword.cs] (20 วินาที ทั้ง T ข้อมูลทดสอบ (1 ไฟล์))

หนังสือพิมพ์แห่งหนึ่งมีเกม crossword ให้เล่นเป็นประจำทุกอาทิตย์ซึ่งยังมีรางวัลสำหรับการตอบเกมที่ถูกต้องด้วย พี่ๆ ที่ต้องการเงินรางวัลนี้ เขารู้คำตอบของเกมนี้อยู่ในหนังสือเฉลยเก่าๆ แต่เนื่องจากหนังสือเก่ามาก ลำดับการเรียงของคำตอบ crossword จึงหายไปด้วย พี่ๆ ที่เอ็งเครียดสุดๆ แต่โชคดียังพอมีคำตอบของ crossword นี้ แต่เขาไม่รู้ว่าจะนำคำใดใส่ในช่องใด จึงมาไหว้วานให้น้องๆ ช่วยพวกพี่ๆ หน่อย (T\_T) ให้เขียนโปรแกรมเพื่อหาการใส่คำตอบลงในตาราง crossword ที่ให้มานี้

ยกตัวอย่างเช่น

มีตาราง crossword ขนาด 5x5

```
#####
#...#
#.#.#
#.#.#
#.#.#
#....
```

และมีคำศัพท์ที่เป็นคำตอบของ crossword คือคำว่า eat, eng, gang

เมื่อนำคำไปใส่ในตาราง crossword จะได้

```
#####
#eng#
#a###
#t###
#gang
```

การอ่านคำในตารางจะอ่านจากบนลงล่าง และซ้ายไปขวาเท่านั้น (ไม่มีการอ่านกลับด้านจากขวาไปซ้าย หรือ อ่านจากล่างขึ้นบน) แต่ละคำอาจมีตัวเชื่อมกันหรือไม่มีเลยก็ได้ แต่หลังจากใส่คำศัพท์ลงในตารางเสร็จแล้ว เมื่ออ่านจากซ้ายไปขวา และบนลงล่างรวมกันต้องพบคำศัพท์ครบทั้ง K คำ และต้องไม่เหลือช่องใส่คำศัพท์ในตารางอีก

ให้นิสิตเขียนโปรแกรมโดยรับตารางและคำศัพท์ ให้พิมพ์ตารางที่ใส่คำศัพท์นี้เสร็จแล้ว (รับประกันว่าตารางสามารถใส่คำศัพท์ที่ให้มีได้ครบแน่ๆ และรับประกันอีกว่ารูปแบบการใส่คำศัพท์จะใส่ได้แค่รูปแบบเดียวเท่านั้น)

### ข้อมูลนำเข้า

บรรทัดแรกรับจำนวนเต็ม T แทนจำนวนข้อมูลทดสอบ

มีทั้งหมด T ข้อมูลทดสอบ แต่ละข้อมูลทดสอบรับ

N M K แทนจำนวนแถว, จำนวนหลักของตาราง และจำนวนคำศัพท์

รับตารางขนาด NxM

อีก K บรรทัดแต่ละบรรทัดรับคำศัพท์

### ข้อมูลส่งออก

แต่ละข้อมูลทดสอบ แสดงตารางที่ใส่คำศัพท์ crossword เสร็จแล้ว และหลังจากพิมพ์เสร็จแล้วให้พิมพ์ --- จบในแต่ละข้อมูลทดสอบด้วย

### ข้อกำหนดของข้อมูลทดสอบ

$1 \leq T \leq 20$

$1 \leq N, M, K \leq 30$

ความยาวแต่ละคำศัพท์ไม่เกิน 30 ตัวอักษร และเป็นตัวพิมพ์ใหญ่ทั้งหมด

ตัวอย่างผลลัพธ์การทำงานของโปรแกรม

Input:

Output:

```
2
10 10 2
#####
#.....#
###.#####
###.#####
###.#####
###.#####
###.#####
###.#####
###.#####
```

```
#####
#MANDRAKE#
#####A#####
#####B#####
#####M#####
#####C#####
#####D#####
#####E#####
#####
```

##### ##### DABMCDE MANDRAKE 7 9 4 #.....## ##### ####.#### ####.#### ###.#### #...#### ##### COMPRO IS VERY EASY	--- #COMPRO## ##### ####V#### ####E#### ###IR#### #EASY#### ##### ---
--	---

- จะพิมพ์คำตอบก่อนที่จะข้อมูลทดสอบ หรือพิมพ์ทีเดียวเลย ทำได้ทั้งคู่
- นับตามข้อมูลทดสอบที่ถูกต้องนี้สึดต้องรับ input และพิมพ์ผลลัพธ์ให้เหมือนกับตัวอย่างข้างบนเท่านั้น

Download ข้อมูลทดสอบได้ที่

<https://drive.google.com/drive/folders/0B0Wd5pG2Psw-QWtib0t2cHNXNEU?usp=sharing>

(ถ้ามี test case ไหนที่คำตอบไม่เหมือนเฉลยแต่ถูกเหมือนกัน โปรดแจ้งและขอภัยครับ T\_T อาจคิด test case ไม่ได้พอ)

----- ☺☺☺ May the force be with you, since you need it a lot ☺☺☺ -----