

หน่วยที่ 2 การทำซ้ำ (Loops)

คอมพิวเตอร์มีความสามารถในการทำงานซ้ำ ๆ ได้ดี การทำซ้ำจะเรียกว่าลูป ซึ่งมีอยู่หลายประเภทโดยการทำลูปนั้นจะต้องมีการตรวจสอบเงื่อนไขด้วยว่าจะให้ทำซ้ำกี่ขั้นตอน เมื่อใดต้องการหยุดทำซ้ำ ในภาษาซีมีคำสั่งลูปการทำซ้ำอยู่สามประเภท ซึ่งจะต้องทำความเข้าใจในการใช้งานคำสั่งแต่ละคำสั่งก่อนจึงจะสามารถนำมาใช้ได้

โดยทั่วไปแล้ว การทำงานของโปรแกรมคอมพิวเตอร์จะทำงานเรียงตามลำดับ ตั้งแต่คำสั่งแรกจนถึงคำสั่งสุดท้าย แต่เราสามารถให้คอมพิวเตอร์ทำงานซ้ำ ๆ ที่คำสั่งชุดหนึ่งได้ โดยใช้คำสั่งควบคุมให้ทำงานซ้ำได้ บางครั้งเรียกว่า คำสั่งลูป ในภาษาซีมีคำสั่งให้ทำงานซ้ำอยู่ 3 รูปแบบ ขึ้นอยู่กับเงื่อนไขของการทำซ้ำ คือ

1. การทำซ้ำด้วยคำสั่ง while
2. การทำซ้ำด้วยคำสั่ง do...while
3. การทำซ้ำด้วยคำสั่ง for

1. การทำซ้ำด้วยคำสั่ง while

คำสั่ง while ใช้สำหรับเขียนโปรแกรมเพื่อให้วนลูปทำงานเช่นเดียวกับคำสั่ง for แต่จะต่างกันตรงที่ลูป while จะไม่รู้จำนวนรอบที่ลูปจะวนทำงานได้ โดยลูปจะทำงานไปเรื่อย ๆ ตราบใดที่เงื่อนไขยังคงเป็นจริงอยู่ จะหยุดก็ต่อเมื่อเงื่อนไขที่ตั้งไว้เป็นเท็จ รูปแบบการทำงานของลูป while จะเริ่มจากการตรวจสอบเงื่อนไข ถ้าผลออกมาเป็นจริงจึงจะทำงานคำสั่งที่อยู่ในลูป

รูปแบบคำสั่ง

```
while ( condition )
    statement;
```

หรือ

```
while ( condition )
{
    statement_1;
    statement_2;
    ...
    statement_n;
}
```

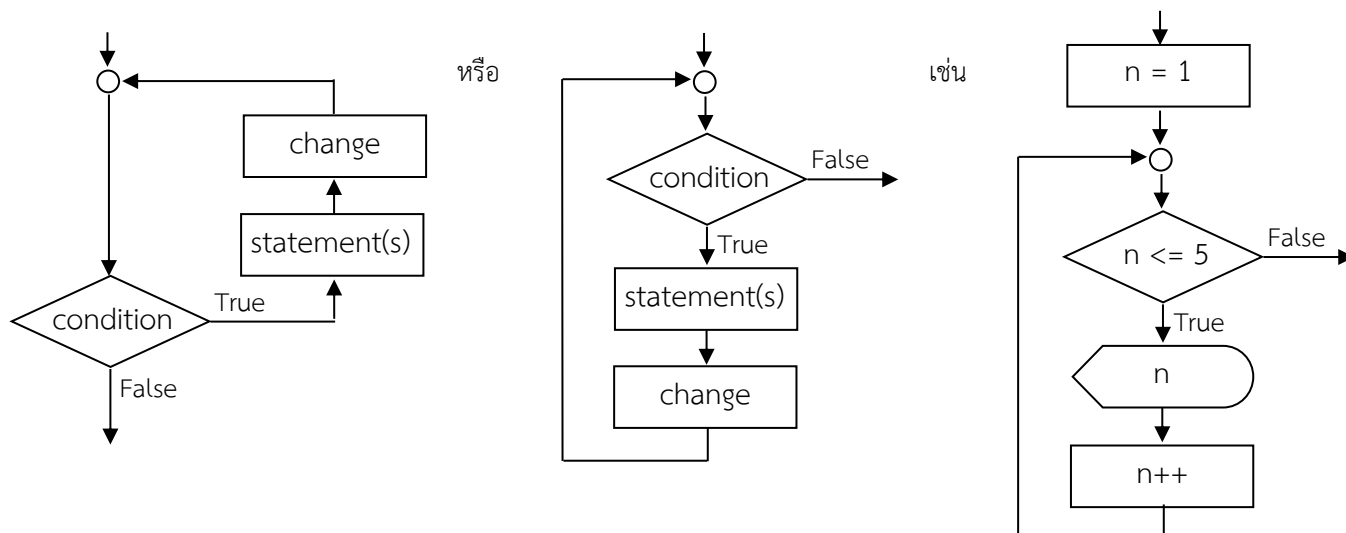
ในขณะที่.....ทำ
หรือ ตราบใดที่.....ทำ

คำอธิบาย

condition คือ **เงื่อนไขการทำซ้ำ** เป็นนิพจน์ใด ๆ ที่สามารถประเมินค่าได้ โดยจะต้องเขียนไว้อยู่ระหว่างเครื่องหมายวงเล็บ () เสมอ เพื่อให้โปรแกรมวนลูปทำงานซ้ำหลังคำสั่ง while ซึ่งเป็นนิพจน์ที่ใช้สำหรับประเมินว่าคำสั่ง (statements) จะถูกประมวลผลหรือไม่ นั่นคือ ถ้าเงื่อนไขการทำซ้ำมีค่าเป็น**จริง (True)** คำสั่ง (statements) จะถูกประมวลผล และถ้าเงื่อนไขการทำซ้ำมีค่าเป็น**เท็จ (False)** คำสั่ง while จะสิ้นสุดลง เช่น $n \geq 0$, $number \leq -5$, $a > b$ เป็นต้น

statement(s) คือ **คำสั่งที่อยู่ภายใต้คำสั่ง while** ที่จะทำให้ทำการวนรอบทำงานซ้ำ ในกรณีที่เงื่อนไขเป็นจริง (True) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในเครื่องหมาย { }

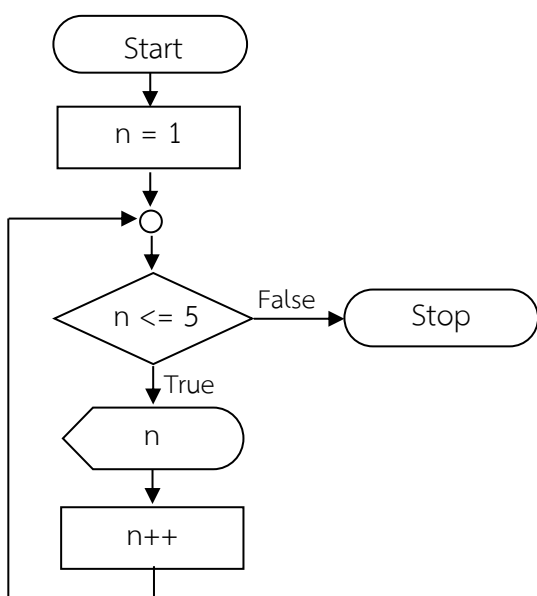
ผังงานโปรแกรมการวนรอบทำซ้ำด้วยคำสั่ง while



**** สิ่งที่ต้องระวัง** ในการใช้คำสั่งการวนรอบทำซ้ำด้วยคำสั่ง while ก็คือ ภายในลูปจะต้องมีหนึ่งคำสั่ง ในการเพิ่ม ลด หรือเปลี่ยนแปลงค่าของตัวแปรที่ใช้ตรวจสอบเงื่อนไข เพื่อให้เงื่อนไขเข้าใกล้จุดที่จะเป็นเท็จ ลูปจะได้มีทางจบงาน ไม่เช่นนั้น จะกลายเป็นลูปที่ไม่มีวันจบ โปรแกรมจะวนทำงานไม่หยุด และไม่สามารถทำงานในคำสั่งต่อจากลูป while ได้

ตัวอย่างโปรแกรมคำสั่งการทำซ้ำด้วยคำสั่ง while

Ex1. โปรแกรมแสดงตัวเลข 1 ถึง 5



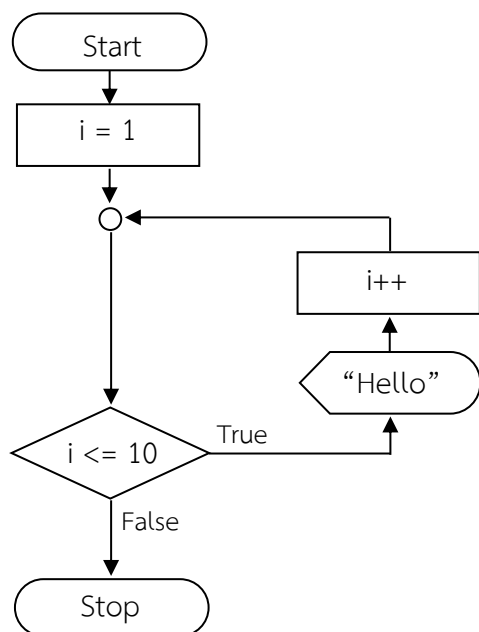
```
#include <stdio.h>
int main() {
    int n=1;
    while ( n<=5 )
    {
        printf ( "%d ", n );
        n++;
    }
    return 0;
}
```

Output
1 2 3 4 5

อัลกอริทึม หรือขั้นตอนวิธี

1. เริ่มต้นทำงาน
2. กำหนดค่าให้ n เท่ากับ 1
3. ในขณะที่ n น้อยกว่าหรือเท่ากับ 5 ทำ
 - 3.1 แสดงค่าของ n
 - 3.2 คำนวณหรือเพิ่มค่าของ n เท่ากับ n + 1
4. จบการทำงาน

Ex2. โปรแกรมแสดงข้อความคำว่า “Hello” จำนวน 10 บรรทัด



```

#include <stdio.h>
int main() {
    int i=1;
    while ( i<=10 )
    {
        printf ( "Hello\n" );
        i++;
    }
    return 0;
}
  
```

Output

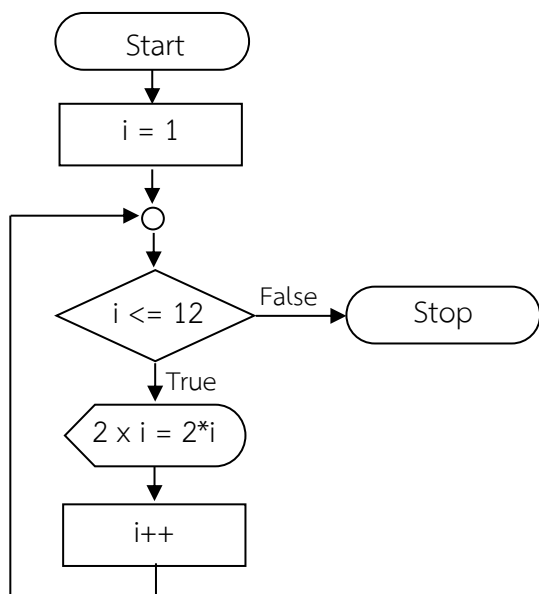
```

Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
  
```

อัลกอริทึม หรือขั้นตอนวิธี

1. เริ่มต้นทำงาน
2. กำหนดค่าให้ i เท่ากับ 1
3. ในขณะที่ i น้อยกว่าหรือเท่ากับ 10 ทำ
 - 3.1 แสดงข้อความคำว่า “Hello”
 - 3.2 คำนวณหรือเพิ่มค่าของ i เท่ากับ $i + 1$
4. จบการทำงาน

Ex3. โปรแกรมแสดงสูตรคูณแม่ 2



```

#include <stdio.h>
int main() {
    int i=1;
    while ( i<=12 )
    {
        printf ( "%d x %d = %d\n", 2, i, 2*i );
        i++;
    }
    return 0;
}

```

Output

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24

```

อัลกอริทึม หรือขั้นตอนวิธี

1. เริ่มต้นทำงาน
2. กำหนดค่าให้ i เท่ากับ 1
3. ในขณะที่ i น้อยกว่าหรือเท่ากับ 12 ทำ
 - 3.1 แสดงผล $2 \times i$ เท่ากับ $2 * i$
 - 3.2 คำนวณหรือเพิ่มค่าของ i เท่ากับ $i + 1$
4. จบการทำงาน

**** ตัวอย่างโปรแกรมที่มีการทำซ้ำแบบไม่มีวันจบ ****

```

#include <stdio.h>
int main() {
    int n=1;
    while ( n<=5 )
    {
        printf ( "Number %d ", n );
        n--;
    }
    return 0;
}

```

Output

```

Number 1
Number 0
Number -1
Number -2
Number -3
Number -4
Number -5
Number -6
Number -7
Number -8
Number -9
Number -10
Number -11
Number -12
Number -13

```

.....

.....

2. การทำซ้ำด้วยคำสั่ง do...while

การทำงานของคำสั่งวนรูปแบบ do...while จะเริ่มจากการทำงานตามคำสั่งของ do ก่อนหนึ่งรอบ แล้วจึงมีการตรวจสอบเงื่อนไขการทำซ้ำที่คำสั่ง while ในส่วนท้ายของลูป ถ้าเงื่อนไขเป็นจริงจึงจะวนกลับไปทำงานตามคำสั่งของ do ในรอบต่อไป จนกว่าเงื่อนไขจะเป็นเท็จ ดังนั้น สรุปได้ว่า การวนรูปแบบ do...while อย่างน้อยจะมีความทำงานตามคำสั่งของ do 1 รอบแน่นอน

รูปแบบคำสั่ง

```
do
    statement;
while ( condition ) ;
```

หรือ

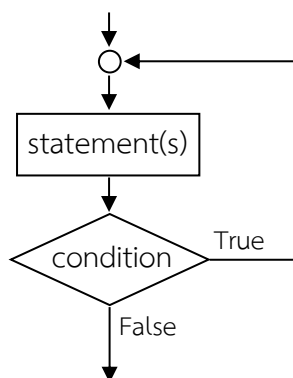
```
do
{
    statement_1;
    statement_2;
    ...
    statement_n;
}
while ( condition ) ;
```

คำอธิบาย

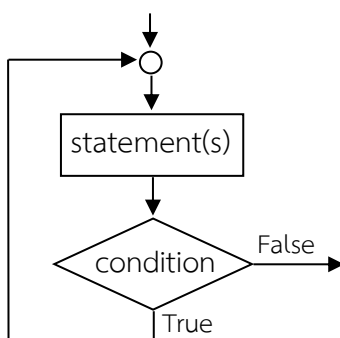
statement(s) คือ คำสั่งที่จะถูกประมวลผลโดยไม่มีเงื่อนไขใด ๆ ในรอบแรก จากนั้นเงื่อนไขการทำซ้ำจะถูกประเมินค่า และคำสั่ง (statements) จะถูกประมวลผลทำงานซ้ำไปทุกครั้งที่เงื่อนไขการทำซ้ำมีค่าเป็นจริง (True) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในเครื่องหมาย { } และในครั้งแรกที่เงื่อนไขการทำซ้ำมีค่าเป็นเท็จ (False) คำสั่ง do...while จะสิ้นสุดลงทันทีหลังจากทำคำสั่ง 1 รอบแล้ว

condition คือ เงื่อนไขการทำซ้ำ เป็นนิพจน์ใด ๆ ที่สามารถประเมินค่าได้ โดยจะต้องเขียนไว้อยู่ระหว่างเครื่องหมายวงเล็บ () เสมอและมีเครื่องหมายเซมิโคลอน (;) ต่อท้าย เพื่อให้โปรแกรมวนลูปทำงานซ้ำหลังคำสั่ง do ซึ่งเป็นนิพจน์ที่ใช้สำหรับประเมินว่าคำสั่ง (statements) จะถูกประมวลผลหรือไม่ นั่นคือ ถ้าเงื่อนไขการทำซ้ำมีค่าเป็นจริง (True) คำสั่ง (statements) จะถูกประมวลผล และถ้าเงื่อนไขการทำซ้ำมีค่าเป็นเท็จ (False) คำสั่ง do...while จะสิ้นสุดลง เช่น $n \geq 0$, $number \leq -5$, $a > b$ เป็นต้น

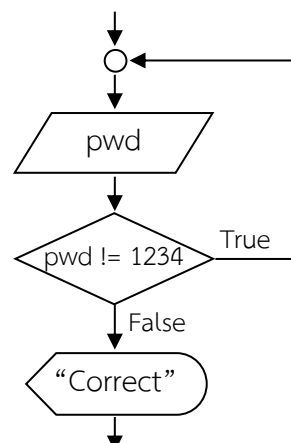
ผังงานโปรแกรมการทำซ้ำด้วยคำสั่ง do...while



หรือ

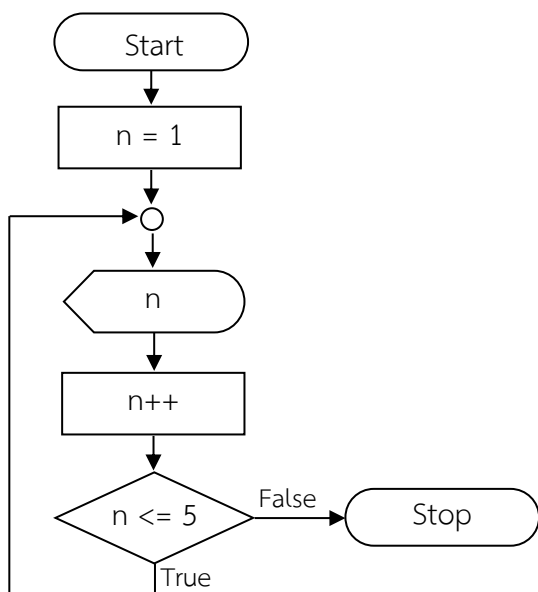


เช่น



ตัวอย่างโปรแกรมคำสั่งการทำซ้ำด้วยคำสั่ง do...while

Ex4. โปรแกรมแสดงตัวเลข 1 ถึง 5

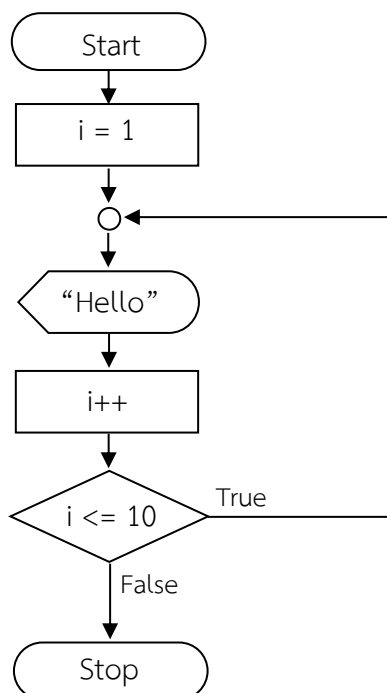


```
#include <stdio.h>
int main() {
    int n=1;
    do
    {
        printf ( "%d ", n );
        n++;
    }
    while ( n<=5 );
    return 0;
}
```

Output

1 2 3 4 5

Ex5. โปรแกรมแสดงข้อความคำว่า “Hello” จำนวน 10 บรรทัด

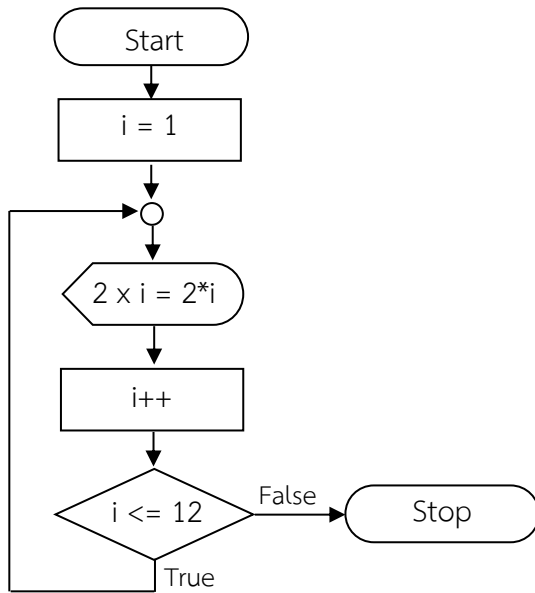


```
#include <stdio.h>
int main() {
    int i=1;
    do
    {
        printf ( "Hello\n" );
        i++;
    }
    while ( i<=10 );
    return 0;
}
```

Output

Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello

Ex6. โปรแกรมแสดงสูตรคูณแม่ 2



```

#include <stdio.h>
int main() {
    int i=1;
    do
    {
        printf ( "%d x %d = %d\n", 2, i, 2*i );
        i++;
    }
    while ( i<=12 );
    return 0;
}

```

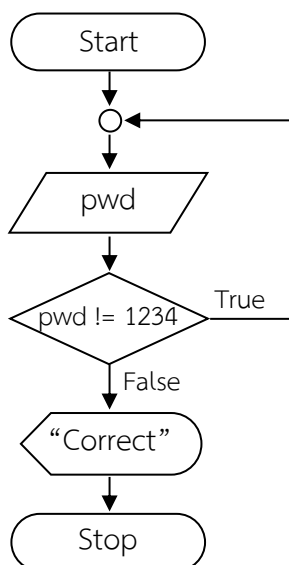
Output

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24

```

Ex7. โปรแกรมรับค่ารหัสผ่านด้วยตัวเลขจำนวนเต็ม 4 หลัก ถ้ารหัสผ่านถูกต้องให้แสดงคำว่า “Correct” แต่ถ้ารหัสผ่านไม่ถูกต้องให้รับค่ารหัสผ่านไปเรื่อย ๆ จนกว่าจะใส่รหัสผ่านถูกต้อง (กำหนดรหัสผ่าน = 1234)



```

#include <stdio.h>
int main() {
    int pwd;
    do
    {
        scanf ( "%d", &pwd );
    }
    while ( pwd != 1234 );
    printf ( "Correct" );
    return 0;
}

```

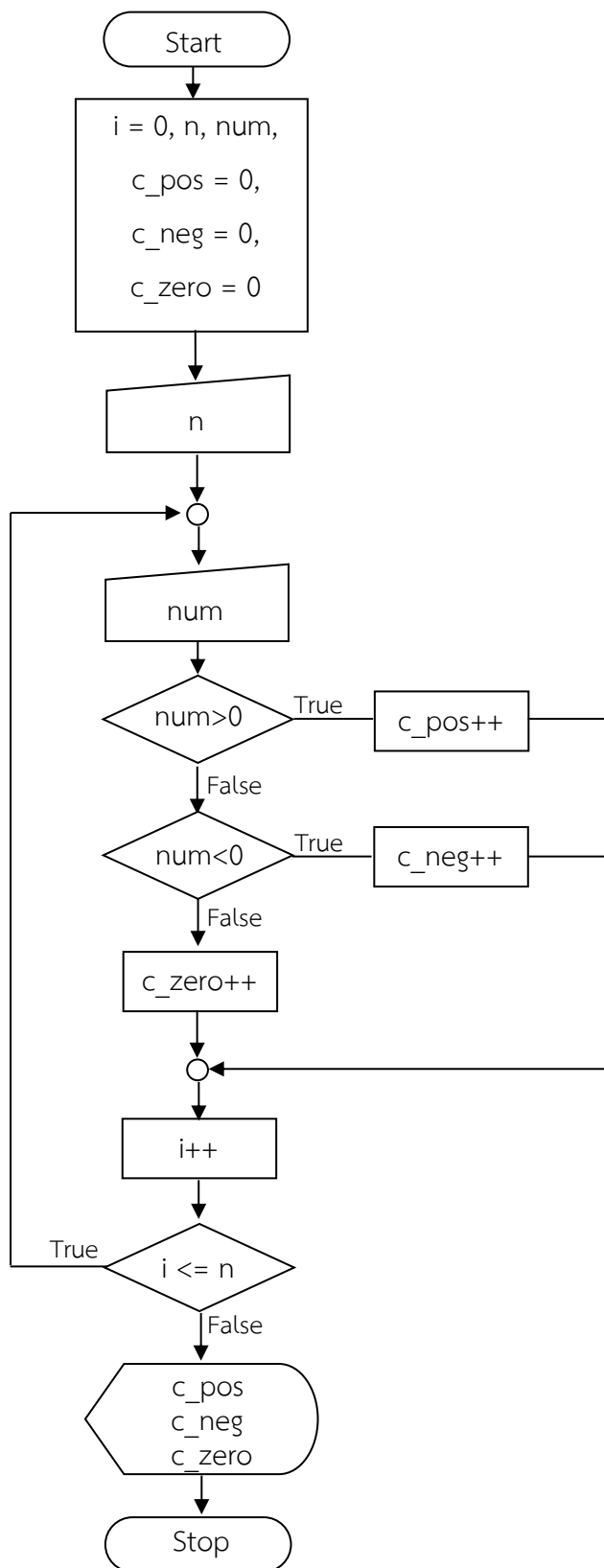
Output

```

1111
1222
1333
1456
7788
1234
Correct

```

Ex8. รับค่าตัวเลขจำนวนเต็ม n จำนวน แล้วกรอกจำนวนเต็มตามจำนวน n จำนวน แล้วนับตัวเลขที่รับเข้ามาว่าเป็นตัวเลขจำนวนเต็มบวก จำนวนที่ติดลบ และเลขศูนย์ มีอย่างละกี่จำนวน



```

#include <stdio.h>
int main() {
    int i=1, n, num, c_pos=0, c_neg=0, c_zero=0;
    printf("n : ");
    scanf("%d", &n);
    do
    {
        printf("Enter number %d : ", i);
        scanf("%d", &num);
        if(num>0)
            c_pos++;
        else if (num<0)
            c_neg++;
        else
            c_zero++;

        i++;
    }
    while(i<=n);
    printf("\nPositive number : %d\n", c_pos);
    printf("Negative number : %d\n", c_neg);
    printf("Zero number : %d", c_zero);
    return 0;
}
  
```

Output

```

n : 12
Enter number 1 : 15
Enter number 2 : 20
Enter number 3 : -22
Enter number 4 : 10
Enter number 5 : -5
Enter number 6 : 0
Enter number 7 : 256
Enter number 8 : 12
Enter number 9 : -8
Enter number 10 : 3
Enter number 11 : -56
Enter number 12 : 0
  
```

```

Positive number : 6
Negative number : 4
Zero number : 2
  
```

3. การทำซ้ำด้วยคำสั่ง for

เหมาะสำหรับกรณีที่เราจำนวนรอบแน่นอนแล้วว่า ต้องการให้วนทำงานตามคำสั่งที่รอบ

รูปแบบคำสั่ง

**for (initial; condition; change)
statement ;**

หรือ

**for (initial; condition; change)
{
statement_1;
statement_2;
...
statement_n;
}**

คำอธิบาย

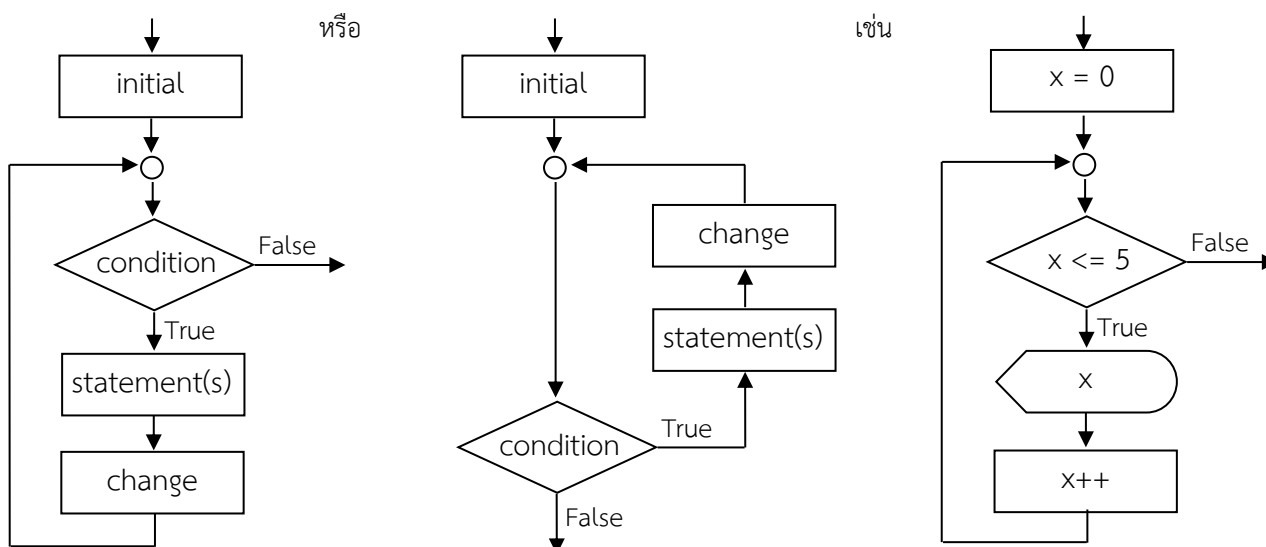
initial คือ การกำหนดค่าเริ่มต้นของตัวนับ เป็นนิพจน์ที่ใช้สำหรับกำหนดค่าเริ่มต้น ให้กับตัวแปรที่ทำหน้าที่ควบคุมการทำงานซ้ำ และจะประมวลผลเพียงครั้งเดียว ถ้ามีค่าเริ่มต้นหลายค่าให้ใช้คอมม่าคั่น (,) ตัวนับ เช่น `n = 0; number = -5; a = b; c = a + b; i = 0 , j = 0; x = 1 , y = 0; a = 5 , b = 10 , c = 15;`

condition คือ เงื่อนไขการทำงานซ้ำ เพื่อให้โปรแกรมวนทำงานซ้ำหลังคำสั่ง for ซึ่งเป็นนิพจน์ที่ใช้สำหรับประเมินว่าคำสั่ง (statements) จะถูกประมวลผลหรือไม่ นั่นคือ ถ้าเงื่อนไขการทำงานซ้ำมีค่าเป็นจริง (True) คำสั่ง (statements) จะถูกประมวลผล และถ้าเงื่อนไขการทำงานซ้ำมีค่าเป็นเท็จ (False) คำสั่ง for จะสิ้นสุดลง เช่น `n >= 0; number <= -5; a > b;`

change คือ การเปลี่ยนแปลงค่าตัวนับ เป็นนิพจน์ที่จะถูกประมวลผลทุกครั้งหลังจากคำสั่งถูกประมวลผล โดยการเปลี่ยนแปลงค่าตัวนับ จะเป็นนิพจน์ใด ๆ ที่สามารถประเมินค่าได้ ซึ่งอาจเป็นนิพจน์กำหนดค่า นิพจน์เพิ่มค่าหรือนิพจน์ลดค่าของตัวแปรครั้งละหนึ่ง หรือมากกว่านั้นก็ได้ เช่น `a++ number-- a = a + 2 c += 5`

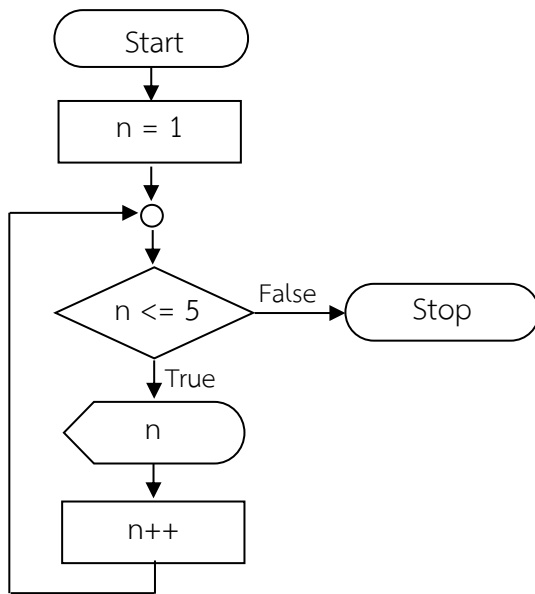
statement(s) คือ คำสั่งที่อยู่ภายใต้คำสั่ง for ที่จะให้ทำการวนรอบทำงานซ้ำ ในกรณีที่เงื่อนไขเป็นจริง (True) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในเครื่องหมาย { }

ผังงานโปรแกรมการทำงานซ้ำด้วยคำสั่ง for



ตัวอย่างโปรแกรมคำสั่งการทำซ้ำด้วยคำสั่ง for

Ex9. โปรแกรมแสดงตัวเลข 1 ถึง 5

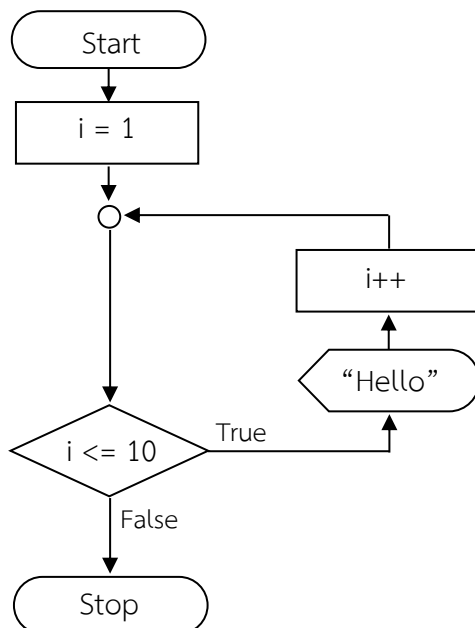


```
#include <stdio.h>
int main() {
    int n;
    for ( n=1; n<=5; n++ )
        printf ( "%d ", n );
    return 0;
}
```

Output

1 2 3 4 5

Ex10. โปรแกรมแสดงข้อความคำว่า “Hello” จำนวน 10 บรรทัด

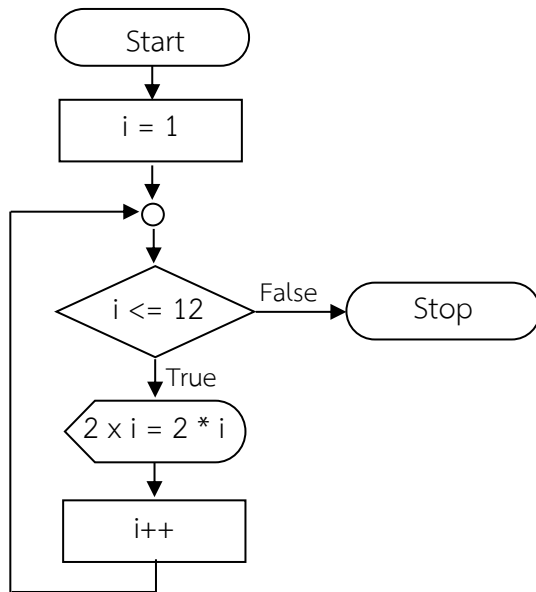


```
#include <stdio.h>
int main() {
    int i;
    for ( i=1; i<=10; i++ )
        printf ( "Hello\n" );
    return 0;
}
```

Output

Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello

Ex11. โปรแกรมแสดงสูตรคูณแม่ 2



```
#include <stdio.h>
int main() {
    int i;
    for ( i=1; i<=12; i++ )
        printf ( "%d x %d = %d\n", 2, i, 2*i );
    return 0;
}
```

Output

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24
```


Ex13. โปรแกรมแสดงคู่อันดับ

```
#include <stdio.h>
int main() {
    int i, j;
    for ( i=1; i<=3; i++ )
    {
        for ( j=1; j<=5; j++ )
            printf ( "(%d,%d) ", i, j );
        printf ( "\n" );
    }
    return 0;
}
```

Output

```
(1,1) (1,2) (1,3) (1,4) (1,5)
(2,1) (2,2) (2,3) (2,4) (2,5)
(3,1) (3,2) (3,3) (3,4) (3,5)
```

จะเห็นว่า ในแต่ละรอบของคำสั่งทำซ้ำซ้อนนอก คำสั่ง printf () ในบรรทัดที่ 7 จะทำงาน 5 ครั้ง และเนื่องจากคำสั่งทำซ้ำซ้อนนอกทำงานทั้งหมด 3 รอบ ดังนั้นคำสั่งในบรรทัดที่ 7 จะทำงานทั้งหมด 15 ครั้ง

Ex14. โปรแกรมแสดงตัวเลข 1 ถึง 5 เป็นรูปสามเหลี่ยม

```
#include <stdio.h>
int main() {
    int i, j;
    for ( i=1; i<=5; i++ )
    {
        for ( j=1; j<=i; j++ )
            printf ( "%d ", j );
        printf ( "\n" );
    }
    return 0;
}
```

Output

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

จะเห็นว่า ในแต่ละรอบของคำสั่งทำซ้ำซ้อนนอก จะทำงาน 5 ครั้ง แต่คำสั่ง printf () ในบรรทัดที่ 7 จะทำงาน i ครั้ง และเนื่องจากคำสั่งทำซ้ำซ้อนนอกทำงานทั้งหมด 5 รอบ ดังนั้นคำสั่งในบรรทัดที่ 7 จะทำงานทั้งหมด 15 ครั้ง

จากตัวอย่าง Ex14. จงเขียนโปรแกรมปรับการแสดงผลรูปสามเหลี่ยมด้วยเครื่องหมาย “*” ดังนี้

แบบชิดซ้าย

แบบชิดขวา

แบบชิดซ้ายกลับหัว

แบบชิดขวากลับหัว

Output

n = 5

```
*
* *
* * *
* * * *
* * * * *
```

Output

n = 5

```
*
* *
* * *
* * * *
* * * * *
```

Output

n = 5

```
* * * * *
* * * *
* * *
* *
*
```

Output

n = 5

```
* * * * *
* * * *
* * *
* *
*
```


goto statement

คำสั่ง **goto** ไม่จำเป็นต้องนำไปใช้กับการทำซ้ำ จะใช้ที่ได้ก็ได้ในโปรแกรม โดยการกำหนดตำแหน่งสำหรับให้กระโดด ตั้งชื่อตำแหน่งแล้วตามด้วยเครื่องหมายโคลอน (Colon) : จากนั้นคำสั่ง **goto** ก็จะทำให้การอ้างอิงตำแหน่งที่ต้องการให้กระโดดไป คำสั่ง **goto** สามารถให้กระโดดไปข้างหน้าหรือย้อนกลับก็ได้

Syntax1	Syntax2
goto label;	label: ←
...	...
...	...
...	...
label: ←	goto label; →
...	...

```
#include <stdio.h>
int printNumbers()
{
    int n = 1;
    label: ←
    printf("%d ", n);
    n++;
    if (n <= 10)
        goto label;
}

int main(){
    printNumbers();
    return 0;
}
```

Output

1 2 3 4 5 6 7 8 9 10

break and continue statement

ทั้งสองคำสั่ง เป็นคำสั่งที่ใช้ในการปรับเปลี่ยนการทำงานแบบปกติของโปรแกรม นำไปใช้ดังนี้

1. คำสั่ง break;

เป็นคำสั่งให้หยุดและออกจากการทำซ้ำ หมายถึง จบออกการทำซ้ำไปของลูปเลย โดยไม่ต้องทำคำสั่งที่อยู่ถัดจาก **break** สามารถใส่คำสั่งไว้ที่ได้ก็ได้ในส่วนที่เป็นคำสั่งที่ต้องการทำซ้ำในคำสั่ง **for**, **while**, **do...while** และ **switch** เมื่อเจอคำสั่ง **break** แล้วลูปจะเลิกการทำงานทันที มักใช้ควบคู่กับคำสั่งควบคุมทางเลือก

2. คำสั่ง continue;

เป็นคำสั่งให้กลับไปตรวจสอบเงื่อนไขของการทำซ้ำใหม่ โดยไม่ต้องทำคำสั่งที่อยู่ถัดจากคำสั่ง **continue** สามารถใส่คำสั่งไว้ที่ได้ก็ได้ในส่วนที่เป็นคำสั่งที่ต้องการทำซ้ำในคำสั่ง **for**, **while**, **do...while** เมื่อสั่งแล้วจะไปเริ่มต้นลูปใหม่อีกครั้งหนึ่งข้ามคำสั่งส่วนที่เหลือของลูปไป มักใช้ควบคู่กับคำสั่งควบคุมทางเลือก

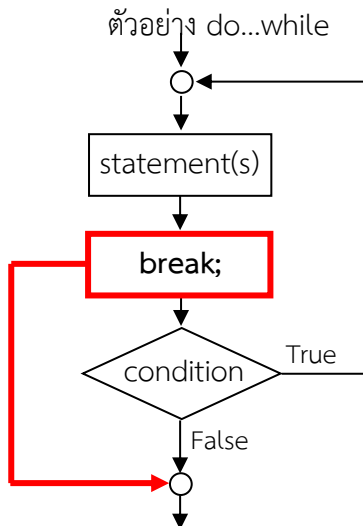
Ex16. การเขียนโปรแกรมทำซ้ำ ที่มีคำสั่งคำสั่ง **break**; และคำสั่ง **continue**;

```
#include <stdio.h>
int main () {
    int n, sum=0;
    float count;
    while(1) // 1 is true
    {
        printf("Enter a number : ");
        scanf("%d", &n);
        if (n<0)
            break;
        count++;
        if (n==0)
            continue;
        sum += n; //sum = sum+n;
    }
    printf("Average = %f", sum/count);
    return 0;
}
```

Output

Enter a number : 17
 Enter a number : 24
 Enter a number : 8
 Enter a number : 0
 Enter a number : 31
 Enter a number : 65
 Enter a number : 0
 Enter a number : 28
 Enter a number : 12
 Enter a number : -5
 Average = 20.555555

ผังงานการหยุดออกจากการทำซ้ำด้วยคำสั่ง break;

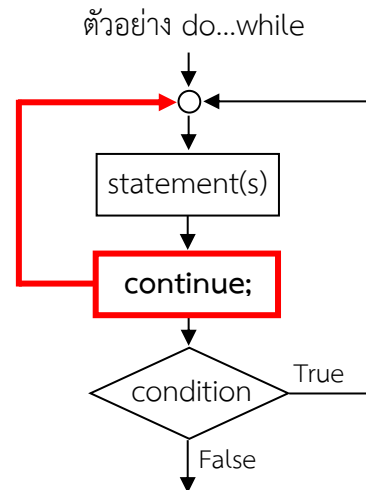


ตัวอย่างโปรแกรม คำสั่ง break;

```

for( init; condition; operation)
{
    // code
    if(condition to break)
    {
        break;
    }
    // code
}
  
```

ผังงานการกลับไปเริ่มทำซ้ำรอบใหม่ด้วยคำสั่ง continue;



ตัวอย่างโปรแกรม คำสั่ง continue;

```

for( init; condition; update)
{
    // ...
    if(condition)
    {
        continue;
    }
    // ...
}
  
```

```

#include <stdio.h>
int main(){
    int i, j;
    // nested for loops with break statement
    // at inner loop
    for (i = 1; i <= 6; ++i) {
        for (j = 1; j <= i; ++j) {
            if (i <= 4) {
                printf("%d ", j);
            }
            else {
                // if i > 4 then this inner most loop will
                // break
                break;
            }
        }
        printf("\n");
    }
    return 0;
}
  
```

Output

```

1
1 2
1 2 3
1 2 3 4
  
```

```

#include <stdio.h>
int main()
{
    int i, j;
    // outer loop with 3 iterations
    for (i = 1; i <= 3; i++) {
        // inner loop to print integer 1 to 4
        for (j = 0; j <= 4; j++) {
            // continue to skip printing number 3
            if (j == 3) {
                continue;
            }
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
  
```

Output

```

0 1 2 4
0 1 2 4
0 1 2 4
  
```

ใบงานหน่วยที่ 2

การทำซ้ำ (Loops)

จงเขียนโปรแกรมแก้โจทย์ปัญหาด้วยคำสั่งการทำซ้ำ (while, do...while, for, การทำซ้ำเชิงซ้อน)

1. จงเขียนโปรแกรมแสดงผลตัวเลขจาก 1 – 20

Output

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

2. จงเขียนโปรแกรมแสดงผลตัวเลขคี่จาก 1 – 20

Output

1 3 5 7 9 11 13 15 17 19

3. จงเขียนโปรแกรมแสดงผลตัวเลขดังนี้

Output

15 14 13 12 11 10

4. จงเขียนโปรแกรมแสดงผลตัวเลข 1 – 50 โดยให้แสดงผลลัพท์บรรทัดละ 10 จำนวน

Output

```
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 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
```

5. จงเขียนโปรแกรมแสดงพยัญชนะภาษาอังกฤษ ตัวพิมพ์ใหญ่ จาก A ถึง Z โดยให้แสดงผลลัพท์

บรรทัดละ 7 จำนวน

Output

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

6. จงเขียนโปรแกรมแสดงพยัญชนะภาษาอังกฤษ ตัวพิมพ์ใหญ่ จาก A ถึง Z โดยให้แสดงผลลัพท์ดังนี้

Output

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

7. จงเขียนโปรแกรมแสดงพยัญชนะภาษาอังกฤษ ตัวพิมพ์เล็ก จาก a ถึง z โดยให้แสดงผลลัพท์ดังนี้

Output

```
      a  b  c  d  e
    f  g  h  i  j  k
  l  m  n  o  p  q  r
s  t  u  v  w  x  y  z
```

8. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก 1 จำนวน (x) แล้วแสดงจำนวนนับที่หารด้วย x ลงตัว ตั้งแต่ 1 ถึง 1000

Input	Output
3	333
7	142
13	76

9. จงเขียนโปรแกรมหาจำนวนนับที่หารด้วย 6 เหลือเศษ 2 และหารด้วย 14 เหลือเศษ 10 ตั้งแต่ 501 ถึง 1000 ให้พิมพ์เพียงว่ามีกี่จำนวน ไม่ใช่พิมพ์ทุกจำนวน

Output

?

10. จงเขียนโปรแกรมหาตัวเลขจำนวนเต็มตั้งแต่ 1 ขึ้นไป ตัวเลขใดที่หาร 21 เหลือเศษ 5 เป็นตัวแรก ให้แสดงตัวเลขจำนวนนั้นออกมา

Output

?

11. จงเขียนโปรแกรมแสดงผลกำลังสองของตัวเลขจาก 1 – 10

Output

1 4 9 16 25 36 49 64 81 100

12. จงเขียนโปรแกรมแสดงผลตัวเลข 1 – 10 และแสดงผลรวมของตัวเลขจาก 1 – 10

Output

1 2 3 4 5 6 7 8 9 10
Sum = 55

13. จงเขียนโปรแกรมรับค่าจำนวนเต็มบวก n หาค่าผลรวมจากสูตรต่อไปนี้

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n}$$

Input	Output
5	Sum = 2.283334
10	Sum = 2.928968
22	Sum = 3.690813

14. จงเขียนโปรแกรมรับค่าแล้วแสดงผลเลขคี่ที่เริ่มจาก 1 จำนวน n ค่า ให้แสดงผลรวมเลขคี่ทั้งหมด (Sum) และค่าเฉลี่ยของเลขคี่ทั้งหมด (Average)

Input	Output
10	1 3 5 7 9 11 13 15 17 19 Sum = 100 Average = 10.000000
5	1 3 5 7 9 Sum = 25 Average = 5.000000

15. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 5 จำนวนที่ไม่เกิน 100 แล้วให้แสดงผลรวม (Sum) และค่าเฉลี่ย (Average) ของเลขจำนวนเต็มทั้งหมด

Input	Output
20 5 15 33 45	Sum = 118 Average = 23.60
28 56 17 80 35	Sum = 216 Average = 43.20

16. จงเขียนโปรแกรมรับตัวเลขจำนวนเต็ม 5 จำนวนที่ไม่เกิน 100 แล้วแสดงผลค่าสูงสุด (Max) และค่าต่ำสุด (Min)

Input	Output
20 5 15 33 45	Max = 45 Min = 5
28 56 17 80 35	Max = 80 Min = 17

17. จงเขียนโปรแกรมในการสร้างค่าดังต่อไปนี้

Output

A	A+3	A+6	A+9
2	5	8	11
4	7	10	13
6	9	12	15
8	11	14	17
10	13	16	19

18. จงเขียนโปรแกรมในการสร้างค่าดังต่อไปนี้

Output

N	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125

19. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก R และ C โดยที่ R และ C คือ จำนวนแถว (Row) และสดมภ์ (Column) ตามลำดับ จากนั้นโปรแกรมจะแสดงตัวเลขจาก 1 จนถึง $R \times C$ เช่น ถ้า $R = 3$ และ $C = 4$ โปรแกรมจะนับตัวเลขจาก 1 ถึง 12 นอกจากนี้ตัวเลขจะถูกพิมพ์ออกมาเป็นแถว ๆ ละ C จำนวน

Input	Output
3 4	1 2 3 4 5 6 7 8 9 10 11 12
4 3	1 2 3 4 5 6 7 8 9 10 11 12
3 2	1 2 3 4 5 6

20. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก n จำนวน แล้วนำมาสร้างสูตรคูณ จากสูตรคูณแม่ 2 ถึงสูตรคูณแม่ n โดยที่ $2 \leq n \leq 10$

Input	Output				
5	2 x 1 = 2	3 x 1 = 3	4 x 1 = 4	5 x 1 = 5	
	2 x 2 = 4	3 x 2 = 6	4 x 2 = 8	5 x 2 = 10	
	2 x 3 = 6	3 x 3 = 9	4 x 3 = 12	5 x 3 = 15	
	2 x 4 = 8	3 x 4 = 12	4 x 4 = 16	5 x 4 = 20	
	2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25	
	2 x 6 = 12	3 x 6 = 18	4 x 6 = 24	5 x 6 = 30	
	2 x 7 = 14	3 x 7 = 21	4 x 7 = 28	5 x 7 = 35	
	2 x 8 = 16	3 x 8 = 24	4 x 8 = 32	5 x 8 = 40	
	2 x 9 = 18	3 x 9 = 27	4 x 9 = 36	5 x 9 = 45	
	2 x 10 = 20	3 x 10 = 30	4 x 10 = 40	5 x 10 = 50	
	2 x 11 = 22	3 x 11 = 33	4 x 11 = 44	5 x 11 = 55	
	2 x 12 = 24	3 x 12 = 36	4 x 12 = 48	5 x 12 = 60	
4	2 x 1 = 2	3 x 1 = 3	4 x 1 = 4		
	2 x 2 = 4	3 x 2 = 6	4 x 2 = 8		
	2 x 3 = 6	3 x 3 = 9	4 x 3 = 12		
	2 x 4 = 8	3 x 4 = 12	4 x 4 = 16		
	2 x 5 = 10	3 x 5 = 15	4 x 5 = 20		
	2 x 6 = 12	3 x 6 = 18	4 x 6 = 24		
	2 x 7 = 14	3 x 7 = 21	4 x 7 = 28		
	2 x 8 = 16	3 x 8 = 24	4 x 8 = 32		
	2 x 9 = 18	3 x 9 = 27	4 x 9 = 36		
	2 x 10 = 20	3 x 10 = 30	4 x 10 = 40		
	2 x 11 = 22	3 x 11 = 33	4 x 11 = 44		
	2 x 12 = 24	3 x 12 = 36	4 x 12 = 48		

21. จงเขียนโปรแกรมแสดงผลตัวเลข 1 – 10 เป็นรูปสามเหลี่ยม

Output

```

1
2 3
4 5 6
7 8 9 10

```

22. จงเขียนโปรแกรมแสดงผลตัวเลข 1 – 10 เป็นรูปสามเหลี่ยม

Output

```

      1
     2 3
    4 5 6
   7 8 9 10
  
```

23. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	1 1 2 1 2 3
5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5

24. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
4	1 2 2 3 3 3 4 4 4 4
2	1 2 2
5	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

25. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
9	1 0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
4	1 0 0 1 1 1 0 0 0 0
2	1 0 0
5	1 0 0 1 1 1 0 0 0 0 1 1 1 1 1

26. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
2	0 1 2 0 1 0
4	0 1 2 3 4 0 1 2 3 0 1 2 0 1 0

27. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	1 1 1 2 2 3
5	1 1 1 1 1 2 2 2 2 3 3 3 4 4 5

28. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	1 2 2 3 3 3
5	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

29. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	3 2 2 1 1 1
5	5 4 4 3 3 3 2 2 2 2 1 1 1 1 1

30. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	1 0 0 1 1 1
5	1 0 0 1 1 1 0 0 0 0 1 1 1 1 1

31. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมพีระมิด

Input	Output
2	<pre> 0 0 1 0 1 2 0 1 0 </pre>
4	<pre> 0 0 1 0 1 2 0 1 2 3 0 1 2 3 4 0 1 2 3 0 1 2 0 1 0 0 </pre>

32. จงเขียนโปรแกรมสร้างตารางการคูณ

Input	Output
7	<pre> 1 1 2 2 4 3 3 6 9 4 4 8 12 16 5 5 10 15 20 25 6 6 12 18 24 30 36 7 7 14 21 28 35 42 49 </pre>
5	<pre> 1 1 2 2 4 3 3 6 9 4 4 8 12 16 5 5 10 15 20 25 </pre>
2	<pre> 1 1 2 2 4 </pre>

33. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มที่มีมากกว่าศูนย์ แล้วคำนวณหาค่าแฟกทอเรียล (factorial) ของตัวเลขจำนวนนั้น (ไม่เกิน 13)

Input	Output
8	40320
6	720
9	362880

34. จงเขียนโปรแกรมรับค่าแล้วคำนวณหาว่าเป็นจำนวนเฉพาะ (Prime number) หรือไม่

Input	Output
17	Prime number.
21	Not prime number.
28	Not prime number.
29	Prime number.
101	Prime number.

35. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมด้วยอักขระ “*”

Input	Output
2	<pre> * * * *</pre>
5	<pre> * * * * * * * * * * * * * * * * * * * * * *</pre>
4	<pre> * * * * * * * * * * * * * * *</pre>

n=4

Row	Blank	*
1	3	1
2	2	3
3	1	5
4	0	7

แนวคิด พิมพ์แถวที่ r

ให้พิมพ์ช่องว่าง (h-r) ตัว ตามด้วยพิมพ์ “*” จำนวน $(2 * r - 1)$

36. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสามเหลี่ยมด้วยอักขระ A – Z

Input	Output
6	<pre> A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J</pre>
2	<pre> A B C D</pre>
5	<pre> A B C D E F G H I J K L M N O P Q R S T U V W X Y</pre>
4	<pre> A B C D E F G H I J K L M N O P</pre>

n=6

Row	Blank	Char
1	5	1
2	4	3
3	3	5
4	2	7
5	1	9
6	0	11

37. จงเขียนโปรแกรมรับค่าแล้วสร้างภูเขารูปสามเหลี่ยมด้วยอักขระ “/” และ “\”

Input	Output
5	<pre> /\ /\ //\ ///\ ////\ /////</pre>
2	<pre> /\ /\</pre>
4	<pre> /\ /\ //\ ///\</pre>

			/	\		
		/	/	\	\	
	/	/	/	\	\	\
/	/	/	/	\	\	\
/	/	/	/	\	\	\

	/	\
/	/	\

		/	\	
	/	/	\	\
/	/	/	\	\
/	/	/	\	\

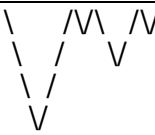
38. จงเขียนโปรแกรมรับค่าแล้วสร้างภูเขารูปสามเหลี่ยมด้วยอักขระ “/” และ “\”

Input	Output
5	
2	
4	

39. จงเขียนโปรแกรมรับค่าแล้วสร้างภูเขารูปสามเหลี่ยมด้วยเครื่องหมาย “/” , “\” และ “.”

Input	Output
5	
2	
4	

40. หินงอก คือ หินที่งอกจากพื้นดิน และ หินย้อย คือ หินที่ย้อยลงมาจากด้านบน เกิดมาโดยเฉพาะภูเขาหินปูน
 จงเขียนโปรแกรมนำอักขระ “\” และ “/” สร้างเป็นรูปหินย้อยตามลักษณะที่กำหนด โดยให้ป้อนตัวเลขค่าแรกเป็นจำนวน
 ของลูกหินย้อย ส่วนค่าที่ตามมาแต่ละบรรทัดเป็นความยาวของหินย้อยแต่ละลูก โดยหินย้อยแต่ละลูกจะนำเชิงเขามาติดกัน
 โดยจำนวนของหินย้อยเป็นจำนวนเต็มที่มีค่าไม่เกิน 20 ส่วนความยาวของหินย้อยเป็นตัวเลขจำนวนเต็มที่มีค่าไม่เกิน 15

Input	Output
3 1 2 3	
4 1 2 1	
2 3 5	

41. จงเขียนโปรแกรมรับค่าแล้วสร้างรูป Diamond ด้วยเครื่องหมาย “*”

Input	Output
3	<pre> * * * * * * * * * * * * *</pre>
5	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *</pre>

41. จงเขียนโปรแกรมรับค่าแล้วสร้างรูป Diamond ด้วยเครื่องหมาย “*”

Input	Output
5	<pre> * * * * * * * * * * * * *</pre>
9	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *</pre>

42. จงเขียนโปรแกรมสร้างรูป 8 เหลี่ยมด้วยอักขระ “*” ตามขนาดตัวเลขที่รับเข้ามา

Input	Output
4	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *</pre>
3	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * *</pre>
2	<pre> * * * * * * * * * *</pre>

43. จงเขียนโปรแกรมสร้างรูปสี่เหลี่ยมจัตุรัสด้วยอักขระ “X”

Input	Output
2	X X X X
5	X X X X X X X X X X X X X X X X

44. จงเขียนโปรแกรมสร้างรูปสี่เหลี่ยมด้วยอักขระ “+” , “*” และ “-”

Input	Output
5	* - - - - + * - - - + + * - - + + + * - + + + + *
9	* - - - - - - - + * - - - - - - + + * - - - - - + + + * - - - - + + + + * - - - + + + + + * - - + + + + + + * - + + + + + + + * - + + + + + + + + *

45. จงเขียนโปรแกรมสร้างรูปสามเหลี่ยมด้วยอักขระ “*”

Input	Output
3	* * * * * * * * * * * * * * * * * * * * * * * * * *
4	* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

46. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปด้วยอักขระ “*”

Input	Output
7	<pre> * * * * * * * * * * * * * * * * * * * * </pre>
5	<pre> * * * * * * * * * * * * * * * * * * </pre>

47. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปสี่เหลี่ยมด้วยอักขระต่าง ๆ ที่กำหนด

Input	Output
* 4	<pre> * * * * * * * * * * * * </pre>
X 5	<pre> X X X X X X X X X X X X X X X X X </pre>
H 6	<pre> H H H H H H H H H H H H H H H H H H H H H H H H </pre>

48. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม แล้วสร้างรูปสามเหลี่ยมตัวเลข

Input	Output
3	<pre> 1 1 2 1 1 2 3 2 1 </pre>
4	<pre> 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 </pre>
7	<pre> 1 1 2 1 1 2 3 3 1 1 2 3 4 3 2 1 1 2 3 4 5 4 3 2 1 1 2 3 4 5 6 5 4 3 2 1 1 2 3 4 5 6 7 6 5 4 3 2 1 </pre>

49. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม แล้วคำนวณหาสามเหลี่ยมปาสคาล (Pascal triangle)

Input	Output
5	<pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 </pre>
7	<pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 </pre>

หมายเหตุ สามเหลี่ยมปาสคาล มีลักษณะคือ มียอดเป็นเลขหนึ่ง และขอบทั้งสองข้างเป็นเลข 1 ตัวเลขที่ปรากฏภายใน เกิดจากการบวกกันของเลขที่อยู่เหนือกว่า ทั้งด้านซ้ายและขวา

50. จงเขียนโปรแกรมรับค่าแล้วแสดงผลตามจำนวนของตัวเลขลำดับฟีโบนัชชี (Fibonacci)

Input	Output
10	0 1 1 2 3 5 8 13 21 34
6	0 1 1 2 3 5
4	0 1 1 2

* ตัวเลขตัวต่อไป เกิดจากการบวกกันของตัวเลข 2 จำนวนก่อนหน้า

* ความสัมพันธ์ $X_n = X_{n-1} + X_{n-2}$ เมื่อ $X_0 = 0, X_1 = 1$

51. การวางเมล็ดข้าวสาร จะวางเรียงลงในช่องตารางเรียงไปเรื่อย ๆ จากช่องที่ 1 เป็นต้นไป โดยช่องที่ 1 วางเมล็ดข้าวสาร จำนวน 1 เมล็ด โดยทุกครั้งที่วางเมล็ดข้าวสารช่องต่อไป จะต้องมีความมากกว่าช่องก่อนหน้า 1 เท่า ตัวอย่าง เช่น ช่องที่ 1 วาง 1 เมล็ด , ช่องที่ 2 วาง 2 เมล็ด , ช่องที่ 3 วาง 4 เมล็ด , ช่องที่ 4 วาง 8 เมล็ด , ตามตาราง

ช่องที่	1	2	3	4	5	6	7	8	9	10	...	...	...	...	...	n-1	n
จำนวนเมล็ดข้าวสาร	1	2	4	8	16	32	64	128	256	512	...	...	...	...	...	...	$(n-1)*2$

จงเขียนโปรแกรมรับค่าจำนวนเต็ม 1 จำนวน แทนจำนวนเมล็ดข้าวสาร แล้วคำนวณหาว่า จะต้องวางข้าวสารช่องที่เท่าไรจึงจะมีค่าเท่ากับจำนวนเต็มนั้น เช่น จำนวนเมล็ดข้าวสาร 100000000 เมล็ด (คำตอบคือ ช่องที่ 28) หรือ จำนวนเมล็ดข้าวสาร 500 เมล็ด (คำตอบคือ ช่องที่ 10)

Input	Output
100000000	28
500	10
26	6
2000	12

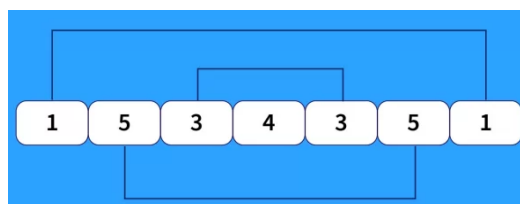
52. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มแล้วแสดงผลตัวเลขกลับกัน

Input	Output
12345	54321
5641	1465
321	123
2564	4652

53. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก 1 จำนวน ไม่เกิน 6 หลัก แล้วทำการตรวจสอบตัวเลขนั้นเป็นเลขพาลินโดรม (Palindrome number) หรือไม่

เลขพาลินโดรม (Palindrome number) คือ ตัวเลขที่สามารถอ่านจากหน้าไปหลังหรือหลังไปหน้าได้แล้วเป็นตัวเลขเดียวกัน หรืออาจเรียกว่า เลขกระจก เช่น 101, 121, 131, 141, 202, 212, 222, 232, 333, 343, 363, 383, 393 เป็นต้น

Input	Output
123	321 is not a palindrome.
101	101 is a palindrome.
12321	12321 is a palindrome.
959	959 is a palindrome.
4567	4537 is not a palindrome.
155	551 is not a palindrome.



Palindrome number.

54. จงเขียนโปรแกรมตรวจสอบตัวเลขจำนวนเต็มบวกตั้งแต่ 1 ถึง 100,000 มีจำนวนเต็มกี่จำนวนที่เป็นทั้งเลขพาลินโดรม (Palindrome number) และเป็นจำนวนเฉพาะ พร้อมแสดงจำนวนเต็มบวกเหล่านั้นด้วย

Output

?
? ? ? ? ? ? ? ? ...

55. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มแล้วแยกตัวประกอบของตัวเลขจำนวนนั้น

Input	Output
128	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$
365	5×73
97	97
12	$2 \times 2 \times 3$

56. จงเขียนโปรแกรมหา หรม. ของตัวเลขจำนวนเต็มบวก 2 จำนวน

Input	Output
105 75	15
255 561	51
96 156	12
140 456	4

57. จงเขียนโปรแกรมหา ครน. ของตัวเลขจำนวนเต็มบวก 2 จำนวน

Input	Output
105 75	30
255 561	255
96 156	24
140 456	32

58. จงเขียนโปรแกรมรับค่าความสูงและความกว้างของบ้านเป็นจำนวนเต็มบวก แล้วสร้างรูปบ้าน โดยความสูงของบ้านสร้างด้วยอักขระ “|” ความกว้างของบ้านสร้างด้วยอักขระ “*” และสร้างหลังคาจั่วด้วยเครื่องหมาย “/” , “\”

Input	Output
1 3	 <pre> / \ / \ /_____\ ***** </pre>
4 2	 <pre> / \ / \ /_____\ ***** ***** ***** ***** </pre>
1 2	 <pre> / \ / \ /_____\ ***** </pre>

59. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก 2 จำนวนแล้วนำไปสร้างปฏิทิน

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นจำนวนเต็มบวก m เป็นข้อมูลจำนวนวันในหนึ่งเดือน โดยกำหนดให้ $28 \leq m \leq 31$
- บรรทัดที่ 2 เป็นจำนวนเต็มบวก w เป็นข้อมูลวันที่ 1 ของเดือนนั้นตรงกับวันใด กำหนดให้ 1 = วันอาทิตย์(Sun), 2 = วันจันทร์(Mon), 3 = วันอังคาร(Tue), 4 = วันพุธ(Wed), 5 = วันพฤหัสบดี(Thu), 6 = วันศุกร์(Fri), 7 = วันเสาร์(Sat) โดยกำหนดให้ $1 \leq w \leq 7$

ข้อมูลส่งออก (Output)

ปฏิทิน จำนวน m วัน และวันที่ 1 ของเดือนนั้นตรงกับ w

Input	Output						
31 1	Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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				
28 2	Sun	Mon	Tue	Wed	Thu	Fri	Sat
		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						
30 5	Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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	

60. เด็กนักเรียนชายโรงเรียนนางรองคนหนึ่ง จะอาบน้ำวันละ 5 ชั้น ถ้าหากเป็นวันพุธที่ตรงกับวันที่เป็นเลขคี่ จะอาบน้ำ 3 ชั้น ถ้าหากเป็นวันศุกร์ตรงกับวันที่เป็นเลขคี่ จะอาบน้ำ 2 ชั้น จงเขียนโปรแกรมคำนวณว่าในหนึ่งเดือน เด็กนักเรียนชายคนนี้จะอาบน้ำกี่ชั้น

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นจำนวนเต็มบวก m เป็นข้อมูลจำนวนวันในหนึ่งเดือน โดยกำหนดให้ $28 \leq m \leq 31$
- บรรทัดที่ 2 เป็นจำนวนเต็มบวก w เป็นข้อมูลวันที่ 1 ของเดือนนั้นตรงกับวันใด กำหนดให้ 1 = วันอาทิตย์(Sun), 2 = วันจันทร์(Mon), 3 = วันอังคาร(Tue), 4 = วันพุธ(Wed), 5 = วันพฤหัสบดี(Thu), 6 = วันศุกร์(Fri), 7 = วันเสาร์(Sat) โดยกำหนดให้ $1 \leq w \leq 7$

ข้อมูลส่งออก (Output)

จำนวนชั้นที่ตักน้ำอาบรวมทั้งหมดในเดือนนั้น

Input	Output
31 1	145
28 2	130
30 5	140

61. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปลูกศรชี้ขึ้น (Arrow Up) ด้วยอักขระต่าง ๆ ดังนี้

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ 1 ตัว ที่จะนำไปสร้างรูปลูกศรชี้ขึ้น

ข้อมูลส่งออก (Output)

แสดงอักขระเป็นรูปลูกศรชี้ขึ้น (Arrow Up) โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

		nth Column						
		0	1	2	3	4	5	6
nth Row	0	.	.	.	C	.	.	.
	1	.	.	C	C	C	.	.
	2	.	C	.	C	.	C	.
	3	C	.	.	C	.	.	C
	4	.	.	.	C	.	.	.
	5	.	.	.	C	.	.	.
	6	.	.	.	C	.	.	.

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ..., $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ..., $2n$ ตามลำดับ อักขระ C อยู่ในสดมภ์ที่ n ทั้งหมด และอักขระ C อยู่ในตำแหน่งที่ $(n-1,1)$, $(n+1,1)$, $(n+2,2)$, ..., $(0,n)$, $(2n,n)$ ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ "." (จุด) ดังตัวอย่าง

$n=3$

	0	1	2	3	4	5	6
0	.	.	.	C	.	.	.
1	.	.	C	C	C	.	.
2	.	C	.	C	.	C	.
3	C	.	.	C	.	.	C
4	.	.	.	C	.	.	.
5	.	.	.	C	.	.	.
6	.	.	.	C	.	.	.

$n=2$

	0	1	2	3	4
0	.	.	C	.	.
1	.	C	C	C	.
2	C	.	C	.	C
3	.	.	C	.	.
4	.	.	C	.	.

$n=4$

	0	1	2	3	4	5	6	7	8
0	.	.	.	.	C	.	.	.	.
1	.	.	.	C	C	C	.	.	.
2	.	.	C	.	C	.	C	.	.
3	.	C	.	.	C	.	.	C	.
4	C	.	.	.	C	.	.	.	C
5	.	.	.	.	C	.	.	.	.
6	.	.	.	.	C	.	.	.	.
7	.	.	.	.	C	.	.	.	.
8	.	.	.	.	C	.	.	.	.

Input	Output
3 C	<pre> . . . C C C C . . . C . C . C . C . . C . . C . . . C C C . . . </pre>
2 X	<pre> . . X . . . X X X . X . X . X . . X X . . </pre>
4 *	<pre> * * * * * . * . * . . * . . * . . * . . * . . . * . . . * * * * * </pre>

62. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปลูกศรชี้ลง (Arrow Down) ด้วยอักขระต่าง ๆ ดังนี้

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ 1 ตัว ที่จะนำไปสร้างรูปลูกศรชี้ลง

ข้อมูลส่งออก (Output)

แสดงอักขระเป็นรูปลูกศรชี้ลง (Arrow Down) โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

n th Row	n th Column						
	0	1	2	3	4	5	6
0	.	.	.	C	.	.	.
1	.	.	.	C	.	.	.
2	.	.	.	C	.	.	.
3	C	.	.	C	.	.	C
4	.	C	.	C	.	C	.
5	.	.	C	C	C	.	.
6	.	.	.	C	.	.	.

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ..., $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ..., $2n$ ตามลำดับ อักขระ C อยู่ในสดมภ์ที่ n ทั้งหมด และอักขระ C อยู่ในตำแหน่งที่ $(n-1, 2n-1)$, $(n+1, 2n-1)$, $(n-2, 2n-2)$, $(n+2, 2n-2)$..., $(0, n)$, $(2n, n)$ ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ "." (จุด) ดังตัวอย่าง

$n=3$

	0	1	2	3	4	5	6
0	.	.	.	C	.	.	.
1	.	.	.	C	.	.	.
2	.	.	.	C	.	.	.
3	C	.	.	C	.	.	C
4	.	C	.	C	.	C	.
5	.	.	C	C	C	.	.
6	.	.	.	C	.	.	.

$n=2$

	0	1	2	3	4
0	.	.	C	.	.
1	.	.	C	.	.
2	C	.	C	.	C
3	.	C	C	C	.
4	.	.	C	.	.

$n=4$

	0	1	2	3	4	5	6	7	8
0	.	.	.	.	C	.	.	.	.
1	.	.	.	.	C	.	.	.	.
2	.	.	.	.	C	.	.	.	.
3	.	.	.	.	C	.	.	.	.
4	C	.	.	.	C	.	.	.	C
5	.	C	.	.	C	.	.	C	.
6	.	.	C	.	C	.	C	.	.
7	.	.	.	C	C	C	.	.	.
8	.	.	.	.	C	.	.	.	.

Input	Output
3 C	<pre> . . . C C C . . . C . . C . . C . C . C . C . . . C C C C . . . </pre>
2 X	<pre> . . X X . . X . X . X . X X X . . . X . . </pre>
4 *	<pre> * * * * . . . * . . . * . * . . * . . * . . . * . * . * * * * * </pre>

63. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปลูกศรชี้ขวา (Arrow Right) ด้วยอักขระต่าง ๆ ดังนี้

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ 1 ตัว ที่จะนำไปสร้างรูปลูกศรชี้ขวา

ข้อมูลส่งออก (Output)

แสดงอักขระเป็นรูปลูกศรชี้ขวา (Arrow Right) โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	.	.	C	.	.
	2	.	.	.	.	.	C	.
	3	C	C	C	C	C	C	C
	4	.	.	.	.	.	C	.
	5	.	.	.	.	C	.	.
	6	.	.	.	C	.	.	.

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ..., $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ..., $2n$ ตามลำดับ อักขระ C อยู่ในแถวที่ n ทั้งหมด และอักขระ C อยู่ในตำแหน่งที่ $(2n-1, n-1)$, $(2n-1, n+1)$, $(2n-2, n-2)$, $(2n-2, n+2)$..., $(n, 0)$, $(n, 2n)$ ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ "." (จุด) ดังตัวอย่าง

$n=3$

	0	1	2	3	4	5	6
0	.	.	.	C	.	.	.
1	.	.	.	.	C	.	.
2	.	.	.	.	.	C	.
3	C	C	C	C	C	C	C
4	.	.	.	.	.	C	.
5	.	.	.	.	C	.	.
6	.	.	.	C	.	.	.

$n=2$

	0	1	2	3	4
0	.	.	C	.	.
1	.	.	.	C	.
2	C	C	C	C	C
3	.	.	.	C	.
4	.	.	C	.	.

$n=4$

	0	1	2	3	4	5	6	7	8
0	.	.	.	.	C	.	.	.	.
1	.	.	.	.	.	C	.	.	.
2	.	.	.	.	.	.	C	.	.
3	.	.	.	.	.	.	.	C	.
4	C	C	C	C	C	C	C	C	C
5	.	.	.	.	.	.	.	C	.
6	.	.	.	.	.	.	C	.	.
7	.	.	.	.	.	C	.	.	.
8	.	.	.	.	C	.	.	.	.

Input	Output
3 C	<pre> . . . C C C . C C C C C C C C C C . . . </pre>
2 X	<pre> . . X X . X X X X X . . . X . . . X . . </pre>
4 *	<pre> * * * . . * * * * * * * * * * * * </pre>

64. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปลูกศรชี้ซ้าย (Arrow Left) ด้วยอักขระต่าง ๆ ดังนี้

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ 1 ตัว ที่จะนำไปสร้างรูปลูกศรชี้ซ้าย

ข้อมูลส่งออก (Output)

แสดงอักขระเป็นรูปลูกศรชี้ซ้าย (Arrow Left) โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	C	.	.	.	.
	2	.	C	.	.	.	.	.
	3	C	C	C	C	C	C	C
	4	.	C	.	.	.	.	.
	5	.	.	C	.	.	.	.
	6	.	.	.	C	.	.	.

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ..., $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ..., $2n$ ตามลำดับ อักขระ C อยู่ในแถวที่ n ทั้งหมด และอักขระ C อยู่ในตำแหน่งที่ $(2n-1, n-1)$, $(2n-1, n+1)$, $(2n-2, n-2)$, $(2n-2, n+2)$..., $(n, 0)$, $(n, 2n)$ ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ “.” (จุด) ดังตัวอย่าง

$n=3$

	0	1	2	3	4	5	6
0	.	.	.	C	.	.	.
1	.	.	C	.	.	.	.
2	.	C	.	.	.	.	.
3	C	C	C	C	C	C	C
4	.	C	.	.	.	.	.
5	.	.	C	.	.	.	.
6	.	.	.	C	.	.	.

$n=2$

	0	1	2	3	4
0	.	.	C	.	.
1	.	C	.	.	.
2	C	C	C	C	C
3	.	C	.	.	.
4	.	.	C	.	.

$n=4$

	0	1	2	3	4	5	6	7	8
0	.	.	.	.	C	.	.	.	.
1	.	.	.	C	.	.	.	.	.
2	.	.	C	.	.	.	.	.	.
3	.	C	.	.	.	.	.	.	.
4	C	C	C	C	C	C	C	C	C
5	.	C	.	.	.	.	.	.	.
6	.	.	C	.	.	.	.	.	.
7	.	.	.	C	.	.	.	.	.
8	.	.	.	.	C	.	.	.	.

Input	Output
3 C	<pre> . . . C C C C C C C C C C . C C C . . . </pre>
2 X	<pre> . . X . . . X . . . X X X X X . X X . . </pre>
4 *	<pre> * * * * * * * * * * * * . * * * * * . . . </pre>

65. จงเขียนโปรแกรมรับค่าแล้วสร้างรูปลูกศร (Arrow) ด้วยอักขระต่าง ๆ ดังนี้

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ 1 ตัว ที่จะนำไปสร้างเป็นรูปลูกศร
- บรรทัดที่ 3 เป็นอักขระ 1 ตัว ที่จะนำไปกำหนดทิศทางของรูปลูกศร ดังนี้

‘U’ หรือ ‘u’ (up) หมายถึง ลูกศรชี้ขึ้น (Arrow Up)

‘D’ หรือ ‘d’ (down) หมายถึง ลูกศรชี้ลง (Arrow Down)

‘R’ หรือ ‘r’ (right) หมายถึง ลูกศรชี้ขวา (Arrow Right) และ

‘L’ หรือ ‘l’ (Left) หมายถึง ลูกศรชี้ซ้าย (Arrow Left)

ข้อมูลส่งออก (Output)

แสดงอักขระเป็นรูปลูกศรชี้ (Arrow) ตามทิศทางต่าง ๆ โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

Arrow Up

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	C	C	C	.	.
	2	.	C	.	C	.	C	.
	3	C	.	.	C	.	.	C
	4	.	.	.	C	.	.	.
	5	.	.	.	C	.	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Down

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	.	C	.	.	.
	2	.	.	.	C	.	.	.
	3	C	.	.	C	.	.	C
	4	.	C	.	C	.	C	.
	5	.	.	C	C	C	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Right

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	.	.	C	.	.
	2	.	.	.	.	.	C	.
	3	C	C	C	C	C	C	C
	4	.	.	.	.	.	C	.
	5	.	.	.	.	C	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Left

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	C	.	.	.	.
	2	.	C	.	.	.	.	.
	3	C	C	C	C	C	C	C
	4	.	C	.	.	.	.	.
	5	.	.	C	.	.	.	.
	6	.	.	.	C	.	.	.

n=3

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ... , $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ... , $2n$ ตามลำดับ อักขระ C อยู่ในสดมภ์ที่ n ทั้งหมด หรือ C อยู่ในแถวที่ n ทั้งหมด และอักขระ C อยู่ตำแหน่งหัวลูกศรตามตัวอย่าง ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ “.” (จุด)

Input	Output
3 V d	<pre> . . . V V V . . . V . . V . . V . V . V . V . . . V V V V . . . </pre>
2 X L	<pre> . . X . . . X . . . X X X X X . X X . . </pre>
2 > r	<pre> . . > > . > > > > > . . . > . . . > . . </pre>
4 # r	<pre> # # # # . # # # # # # # # # # # # # </pre>
3 A u	<pre> . . . A A A A . . . A . A . A . A . . A . . A . . . A A A . . . </pre>

66. จงเขียนโปรแกรมสร้างรูปลูกศร (Arrow) ด้วยอักขระต่าง ๆ ดังนี้ (จาก มทส.)

ข้อมูลนำเข้า (Input)

- บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มบวก n ข้อมูลซึ่งสัมพันธ์กับขนาดของรูปลูกศร โดย $2 \leq n \leq 50$
- บรรทัดที่ 2 เป็นอักขระ C ที่จะนำไปสร้างเป็นรูปลูกศร
- บรรทัดที่ 3 เป็นสายอักขระ ความยาว k โดย $1 \leq k \leq 255$ ซึ่งอักขระในสายอักขระประกอบด้วย ดังนี้
 - ‘U’ (up) หมายถึง ลูกศรชี้ขึ้น (Arrow Up)
 - ‘D’ (down) หมายถึง ลูกศรชี้ลง (Arrow Down)
 - ‘R’ (right) หมายถึง ลูกศรชี้ขวา (Arrow Right) และ
 - ‘L’ (Left) หมายถึง ลูกศรชี้ซ้าย (Arrow Left)

ข้อมูลส่งออก (Output)

มี $2n+1$ บรรทัด แสดงสายอักขระของรูปลูกศร เรียงจากซ้ายไปขวา

โดยรูปลูกศรแต่ละรูปจะเป็นสายอักขระเป็นรูปลูกศรชี้ (Arrow) ตามทิศทางต่าง ๆ โดยมีขนาด $2n+1$ แถว $2n+1$ คอลัมน์ ดังนี้

Arrow Up

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	C	C	C	.	.
	2	.	C	.	C	.	C	.
	3	C	.	.	C	.	.	C
	4	.	.	.	C	.	.	.
	5	.	.	.	C	.	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Down

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	.	C	.	.	.
	2	.	.	.	C	.	.	.
	3	C	.	.	C	.	.	C
	4	.	C	.	C	.	C	.
	5	.	.	C	C	C	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Right

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	.	.	C	.	.
	2	.	.	.	.	.	C	.
	3	C	C	C	C	C	C	C
	4	.	.	.	.	.	C	.
	5	.	.	.	.	C	.	.
	6	.	.	.	C	.	.	.

n=3

Arrow Left

		n th Column						
		0	1	2	3	4	5	6
n th Row	0	.	.	.	C	.	.	.
	1	.	.	C	.	.	.	.
	2	.	C	.	.	.	.	.
	3	C	C	C	C	C	C	C
	4	.	C	.	.	.	.	.
	5	.	.	C	.	.	.	.
	6	.	.	.	C	.	.	.

n=3

ตัวอย่าง ของ $n=3$ หากนับสดมภ์ (Column) ซ้ายสุดเป็นสดมภ์ที่ 0 แล้วสดมภ์ที่อยู่ทางขวามือถัดมาเป็นสดมภ์ที่ 1, 2, 3, ... , $2n$ ตามลำดับ และ แถว (Row) ที่อยู่บนสุดเป็นแถวที่ 0 แล้วแถวด้านล่างลำดับถัดมาเป็นแถวที่ 1, 2, 3, ... , $2n$ ตามลำดับ อักขระ C อยู่ในสดมภ์ที่ n ทั้งหมด หรือ C อยู่ในแถวที่ n ทั้งหมด และอักขระ C อยู่ตำแหน่งหัวลูกศรตามตัวอย่าง ส่วนตำแหน่งอื่น ๆ จะเป็นอักขระ “.” (จุด)

Input	Output
2 X URDD	<pre> . . X X X X . . . X X X X X X . . X . X . X X X X X X X . X . X X . X . X . . X X . . . X X X . . X X X . . . X X X X . . </pre>
2 > R	<pre> . . > > . > > > > > . . . > . . . > . . </pre>
3 * UDLRD	<pre> . . . * * * * * * * * * * * . * . * * * * * . . * . . * * . . . * . . . * * * * * * * * * * * * * * * * * . . . * * * * . . . * * . . . * * * * * . . . * * * * . . . * * . . . * * * * * . . . * * * * . . . * * * * * . . . * * * * . . . * * * * * </pre>

67. จงเขียนโปรแกรมสร้างกรอบซ้อนกรอบด้วยอักขระ “*” ตามขนาดตัวเลขที่รับเข้ามา

Input	Output
1	*
2	<pre> * * * * * * * * * * * * * * * * * </pre>
3	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * </pre>
4	<pre> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * </pre>

68. จงเขียนโปรแกรมรับค่าแล้วสร้างภูเขาเป็นเส้นตรง และประกอบด้วยสามเหลี่ยมหน้าจั่ว จำนวน N ลูก ($1 \leq N \leq 21$) แล้วบันทึกตำแหน่งเริ่มต้น S ($1 \leq S \leq 60$) และความสูงของภูเขา H ($1 \leq H \leq 10$) แต่ละลูกเอาไว้ รูปภูเขาแต่ละลูกประกอบด้วยเนินเขา เครื่องหมาย “/” หรือ “\” และพื้นที่ป่าไม้ “X” โดยความสูงของภูเขาสัมพันธ์กับความสูง ดังตัวอย่าง (จาก มทส.)

$\wedge$ $\wedge$ $\wedge$	$\wedge$ $\wedge$ $\wedge$ $\wedge$	$\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$	$\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$	$\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$ $\wedge$
H = 1	H = 2	H = 3	H = 4	H = 5
มีเฉพาะยอดเขาเท่านั้น ไม่มีพื้นที่ป่าไม้				

รูปแสดงความสัมพันธ์ของความสูงของภูเขาแต่ละระดับ $H = 1 - 5$

จงเขียนโปรแกรมวาดรูปภูเขา โดยมีเงื่อนไขดังนี้

- ไม่มีภูเขาลูกใด ถูกภูเขาอื่นบังมิดทั้งลูก (จำนวนของยอดเขาที่ปรากฏต้องเท่ากับจำนวนของภูเขาที่บันทึกไว้)
- ตำแหน่งที่เนินเขา เครื่องหมาย “/” หรือ “\” ของภูเขาสองลูกเหลื่อมกันให้แทนด้วยตัวอักษรวิหิพิมพ์เล็ก “v”
- ตำแหน่งที่เนินเขาของภูเขาลูกหนึ่ง เหลื่อมกับพื้นที่ป่าไม้ของภูเขาอีกลูกหนึ่งให้ถือว่าเป็นพื้นที่ป่าไม้ ให้แทนด้วยตัวอักษรเอ็กซ์พิมพ์ใหญ่ “X”
- ในการแสดงผลสำหรับพื้นที่ว่างให้แสดงด้วยเครื่องหมายจุด “.” เท่านั้น

ตัวอย่าง เช่น มีภูเขา 3 ลูก ซึ่งมีค่า (S, H) เป็นดังนี้ $(4, 6) (1, 4) (15, 3)$

```

..... /\ .....
..... /XX\ .....
... /\ . /XXXX\ .....
.. /XXvXXXXXXXX\ ... /\ .
. /XXXXXXXXXXXXX\ . /XX\ .
/XXXXXXXXXXXXXXXXXvXXXX\

```

ข้อมูลนำเข้า (Input)

บรรทัดแรก คือ จำนวนลูกของภูเขา (N)

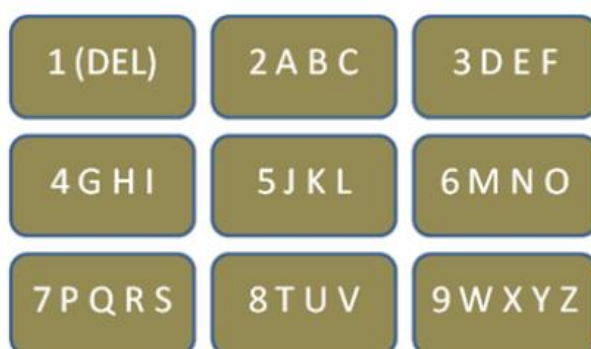
บรรทัดถัดมา จำนวน N บรรทัด แสดงตำแหน่งเริ่มต้น (S) และความสูง (H) ของภูเขาแต่ละลูกคั่นด้วยช่องว่าง 1 ช่อง

ข้อมูลส่งออก (Output)

แสดงรูปภาพภูเขา จำนวน N ลูก โดยความสูงของรูปเท่ากับความสูงของยอดเขาที่สูงที่สุด ตัวอักษรซ้ายสุดของรูปตรงกับตำแหน่ง $S = 1$ และของด้านขวาสุดต้องเป็นส่วนหนึ่งของภูเขาอย่างน้อย 1 ลูก

Input	Output
3 5 6 2 4 16 3	<pre>/\...../XX\...../\./XXXX\..... .../XXvXXXXXXXX\.../\.. .. /XXXXXXXXXXXXX\./XX\.. ./XXXXXXXXXXXXXXXXXvXXXX\ </pre>
5 1 4 6 7 12 6 21 5 41 3	<pre>/\...../XX\./\...../XXXX\XX\...../\.. .../\.../XXXXXXXXXX\.../XX\..... .. /XX\./XXXXXXXXXXXXX\./XXXX\...../\.. ./XXXXvXXXXXXXXXXXXXXXXXvXXXXX\...../XX\.. /XXXXXXXXXXXXXXXXXXXXXXXXXXXXX\...../XXXX\ </pre>

69. กำหนดปุ่มกดโทรศัพท์มือถือเป็นดังนี้ (จาก มทส.)



การกดปุ่มแต่ละครั้งตัวอักษรจะปรากฏวนกันไปตามจำนวนครั้งที่กดตามลำดับ (เฉพาะ ‘A’ – ‘Z’ ไม่มีตัวเลข) ตัวอย่างเช่น การกดปุ่ม 2 จะปรากฏตัวอักษร A B C A B ... วนกันไป ถ้ากดปุ่มนี้จำนวน 5 ครั้งตัวอักษรที่ได้คือ B และถ้ากดปุ่ม 7 จำนวน 2 ครั้งจะได้ตัวอักษร Q เป็นต้น สำหรับปุ่มหมายเลข 1 จะเป็นการลบตัวอักษรที่พิมพ์ไปก่อนหน้านี้ ครั้งละ 1 ตัวอักษร หากไม่มีตัวอักษรเหลืออยู่ในข้อความ การกดปุ่มนี้จะไม่ส่งผลใด ๆ ทั้งสิ้น การเลื่อนนิ้วแต่ละครั้ง (ไปที่ปุ่มใหม่หรือปุ่มเดิมก็ดี) จะนับเริ่มใหม่ที่ตัวอักษรตัวแรกของปุ่มนั้นเสมอ

นางรองสังเกตเห็นนางเอกกำลังกดปุ่มโทรศัพท์ เพื่อส่งข้อความ SMS ในพริบตาแรก นางรองเห็นว่านางเอกเริ่มต้นกดที่ปุ่มไหน ก่อนที่นางเอกจะรู้ว่าถูกแอบมอง จึงปลื้มหลบพ้นสายตา นางรอง อย่างไรก็ตามหลังจากนั้นนางรองก็ยังสามารถสังเกตเห็นได้ว่านางเอกเลื่อนนิ้วไปทางไหน เพื่อกดปุ่มถัดไป เช่น อยู่ทางซ้าย/ขวา หรือ บน/ล่าง เทียบกับปุ่มปัจจุบันไปที่ปุ่มจนกว่านางเอกจะพิมพ์ข้อความเสร็จ ตัวอย่าง เช่น ถ้าครั้งแรก นางเอกกดเลข 8 จำนวน 1 ครั้ง ซึ่งจะได้ตัวอักษร ‘T’ แล้วเลื่อนนิ้วไปทางขวา 1 ปุ่ม ขึ้นบน 1 ปุ่ม และกดปุ่มนั้น 6 ครั้ง แสดงว่านางเอก กดเลข 6 ซึ่งจะได้ตัวอักษร ‘O’ ดังนั้นข้อความที่นางเอกกดอ่านได้เป็น “TO” จากแฟ้มข้อมูลการพิมพ์ SMS ที่นางรองสังเกตเห็น จึงเขียนโปรแกรมหาข้อความที่นางเอกพิมพ์ (นางเอกไม่เลื่อนนิ้วออกนอกแป้นตัวเลขเลย)

ข้อมูลนำเข้า (Input)

บรรทัดแรก คือ จำนวนครั้งที่เลือกกดปุ่ม N ($1 \leq N \leq 80$)

บรรทัดที่สอง ปุ่มแรกที่ถูกกด S ($1 \leq S \leq 9$) และจำนวนครั้งที่กด M ($1 \leq M \leq 4096$) คั่นด้วยช่องว่าง 1 ช่อง

บรรทัดถัดมา จำนวน $N - 1$ บรรทัด แต่ละบรรทัดจะประกอบด้วยตัวเลข 3 จำนวน คือ

- จำนวนแรก แสดงทิศทางแนวนอน H จากปุ่มที่ถูกกดก่อนหน้านี้
- จำนวนที่สอง แสดงทิศทางแนวตั้ง V จากปุ่มที่ถูกกดก่อนหน้านี้
- จำนวนที่สาม แสดงจำนวนครั้งที่กดปุ่ม (M)

โดยจำนวนทั้งสามจะคั่นด้วยช่องว่าง 1 ช่อง

กำหนดค่า

ทิศทางแนวนอน $H \in \{-2, -1, 0, 1, 2\}$

โดยจำนวนลบแสดงการเลื่อนไปทางซ้าย และจำนวนบวกแสดงการเลื่อนไปทางขวา

ทิศทางแนวตั้ง $V \in \{-2, -1, 0, 1, 2\}$

โดยจำนวนลบแสดงการเลื่อนไปขึ้นบน และจำนวนบวกแสดงการเลื่อนไปลงล่าง

ค่าสัมบูรณ์ของตัวเลขแสดงจำนวนปุ่มที่เลื่อนไปในทิศทางนั้น ๆ ดังนั้น เลข 0 หมายถึง ไม่มีการเลื่อนในแนวนั้น

ข้อมูลส่งออก (Output)

ข้อมูลบรรทัดเดียว แสดงข้อความที่พิมพ์(ไม่มีการเว้นช่องว่าง) ถ้าไม่ได้พิมพ์อะไรเลยให้แสดงว่า null

Input	Output
4 5 3 1 0 3 -1 1 3 1 -2 2	LOVE
2 9 6 -2 -2 5	null
5 3 3 0 0 2 -2 0 1 2 1 3 0 1 2	FOX

ASCII Table

Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL (null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH (start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX (start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX (end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT (end of transmission)	36	24	044	0100100	\$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ (enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK (acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL (bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS (backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT (horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF (line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT (vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF (form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR (carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO (shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI (shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE (data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1 (device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2 (device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3 (device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4 (device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK (negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN (synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB (end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN (cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM (end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB (substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC (escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS (file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	
29	1D	035	0011101	GS (group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS (record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US (unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]