



ค่ายโอลิมปิกวิชาการ สอวน. วิชาคอมพิวเตอร์ ค่ายที่ 1  
ปีการศึกษา 2564 ศูนย์โรงเรียนสามเสนวิทยาลัย

ตอนที่ 4 | 21 มีนาคม 2565

---

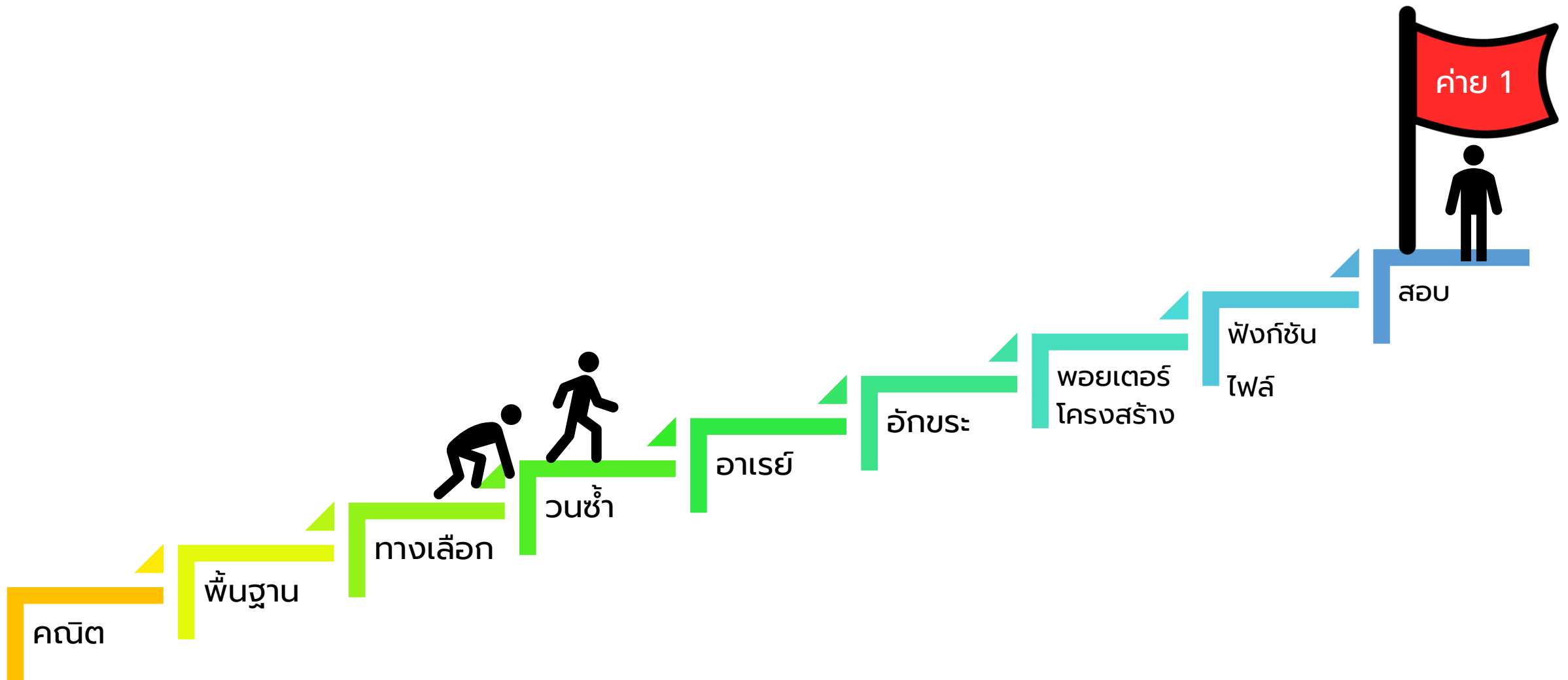
# โปรแกรมประยุกต์ 1

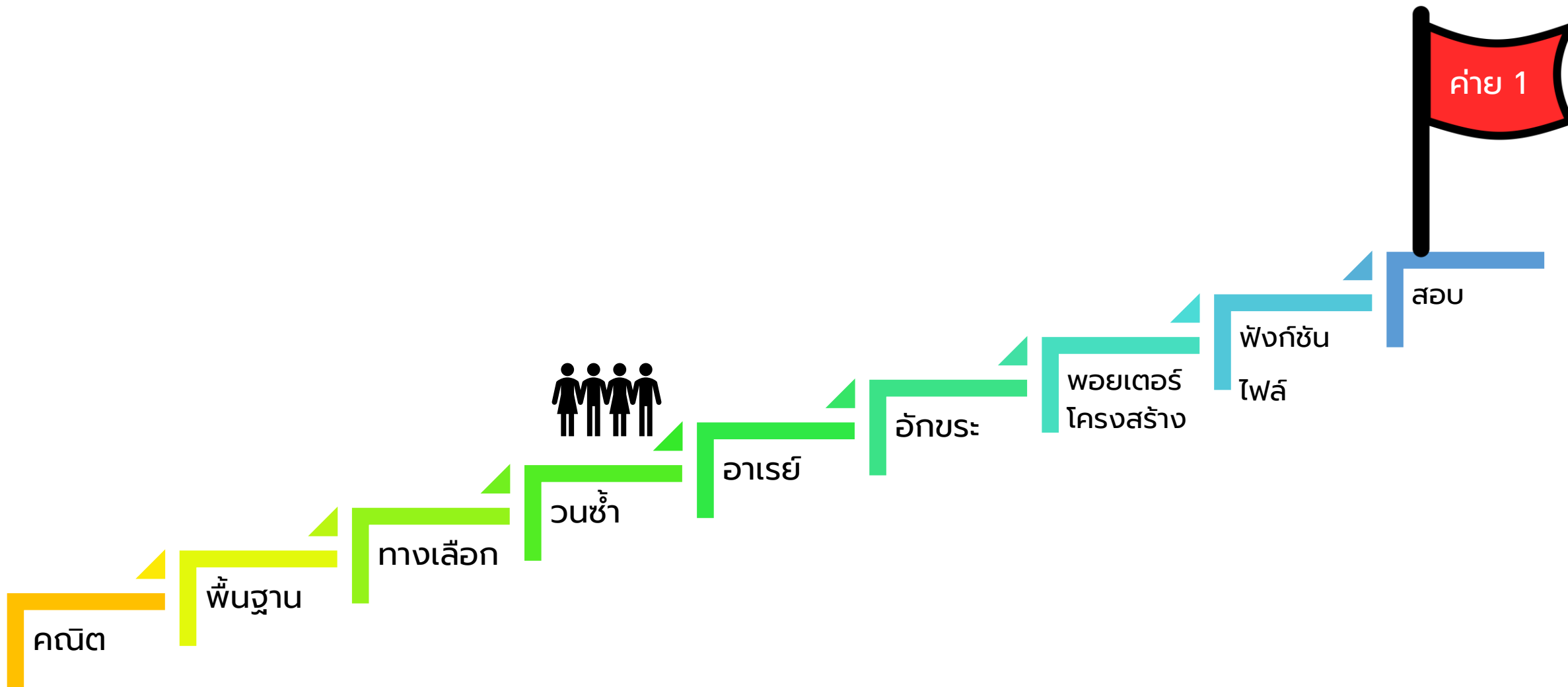


# จุดประสงค์ของการเรียน



อธิบายลักษณะและแนวคิดในการ  
แก้ปัญหของโจทย์ที่ได้รับได้





# ประเภทของตัวแปร



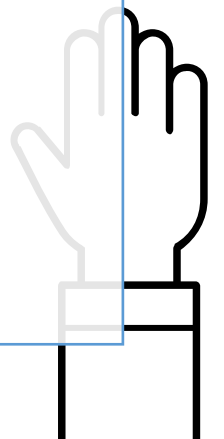
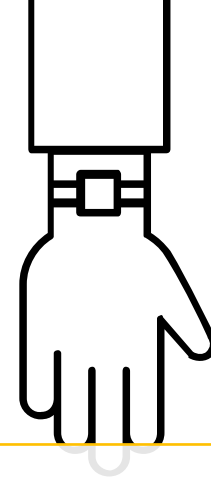
# ประเภทของตัวแปร

## ตัวแปรโลคอล (Local Variable)

- คือ ตัวแปรที่ประกาศภายในฟังก์ชันหลัก หรือการใช้งานได้เฉพาะฟังก์ชันที่ประกาศไว้เท่านั้น

## ตัวแปรโกลบอล (Global Variable)

- คือ ตัวแปรที่ประกาศไว้ภายนอกฟังก์ชันหลัก ซึ่งฟังก์ชันอื่น ๆ ก็สามารถเรียกใช้งานตัวแปรแบบนี้ได้



**Only Programmers Will Understand This**

**Global variable**

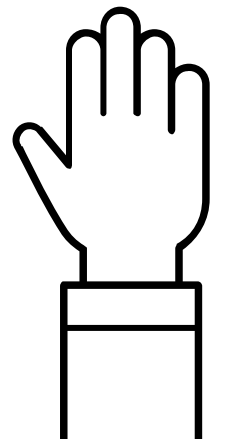
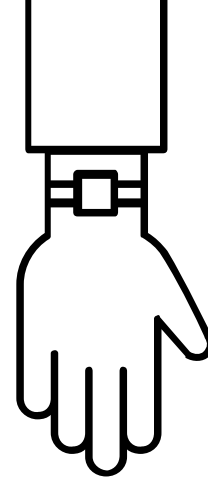
**Local Variables**

**Fb.com/Bookfaceimpact**



# ตัวอย่าง

```
#include<stdio.h>
int x;
int main()    // x เป็นตัวแปร Global
{
    int y;    // y เป็นตัวแปร Local
}
```





# ตัวอย่าง

```
#include<stdio.h>
int x=10;           //x=10 เป็นตัวแปร Global
int main()
{
    int x=20;       //x=20 เป็นตัวแปร Local
    printf("%d",x) ;
}
```

Input-Output

20

# C Scope Rules

ตัวแปร Local Variables นั้นจะถูกจำกัดอยู่ในขอบเขตที่ตัวเองได้เกิดขึ้น

```
#include<stdio.h>
int main()
{
    int i=10;
    for(int i=20;i<=30;i+=5){
        printf("%d ",i);
    }
}
```

Input-Output

20 25 30

# C Scope Rules

ตัวแปร Local Variables นั้นจะ  
ถูกจำกัดอยู่ในขอบเขตที่ตัวเองได้เกิดขึ้น

```
#include<stdio.h>
int main()
{
    int i=10;
    for(int i=20;i<=30;i+=5) {
        printf("%d ",i);
    }
    printf("%d ",i);
}
```

Input-Output

20 25 30 10


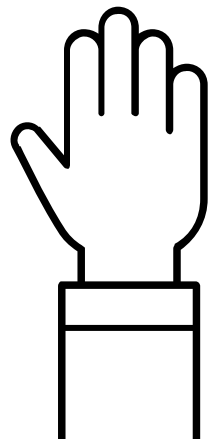
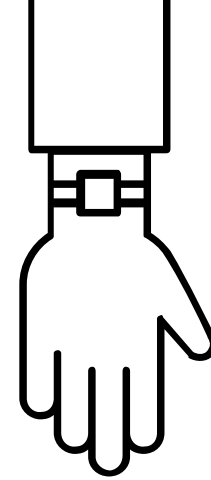
# คำสั่ง Goto



# คำสั่ง goto

สั่งให้ข้ามไปทำงานตามประโยคคำสั่งหรือชื่อที่กำหนดขึ้นโดยไม่มีเงื่อนไขใดๆ มักนิยมใช้ในการออกจาก nested loop

```
goto label;  
... ..  
... ..  
  
label:  
... ..  
... ..
```

A diagram illustrating the 'goto' statement. A line starts from the 'goto label;' statement, goes left, then down, then right, ending with an arrow pointing to the 'label:' block.

# ตัวอย่าง โปรแกรมแสดงเลข 1-5 จำนวน 5 บรรทัด

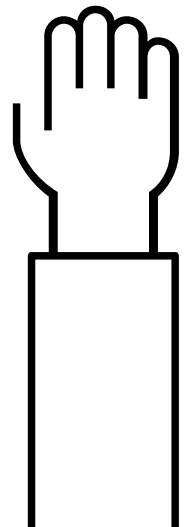
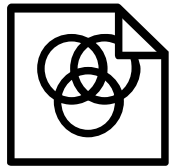
## Input-Output

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
i'm done
```

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

        printf("\n");
    }

    printf("\ni'm done");
    return 0;
}
```



# ตัวอย่าง แก้ไขโปรแกรมเดิมเพื่อให้ แสดงเลข 1-3

## Input-Output

```
1 2 3
i'm done
```

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

# การใช้ Loop ซ้อน Loop (Nested Loop)





# การวิเคราะห์ สำหรับโจทย์ประยุกต์



# ตัวอย่างกระบวนการคิดเพื่อสร้างโปรแกรม

โปรแกรมรับค่าตัวเลขจำนวนเต็มบวก  
แล้วบอกว่าเป็น**จำนวนเฉพาะหรือไม่**

**จำนวนเฉพาะ คืออะไร**

จำนวนเฉพาะหมายถึง จำนวนที่ไม่มีตัวเลขใด  
หารลงตัว นอกจาก 1 และตัวมันเองเท่านั้น เช่น  
2, 3 ,5,7 โดยจำนวนเฉพาะจะเริ่มต้นที่เลข 2

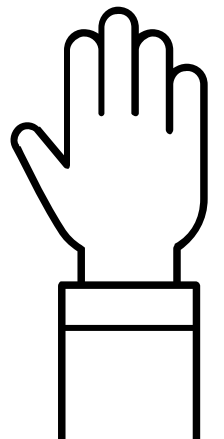
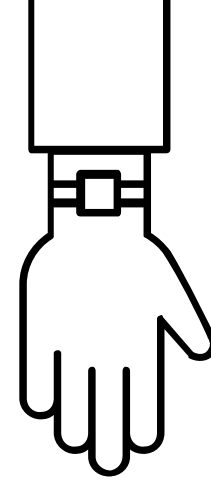
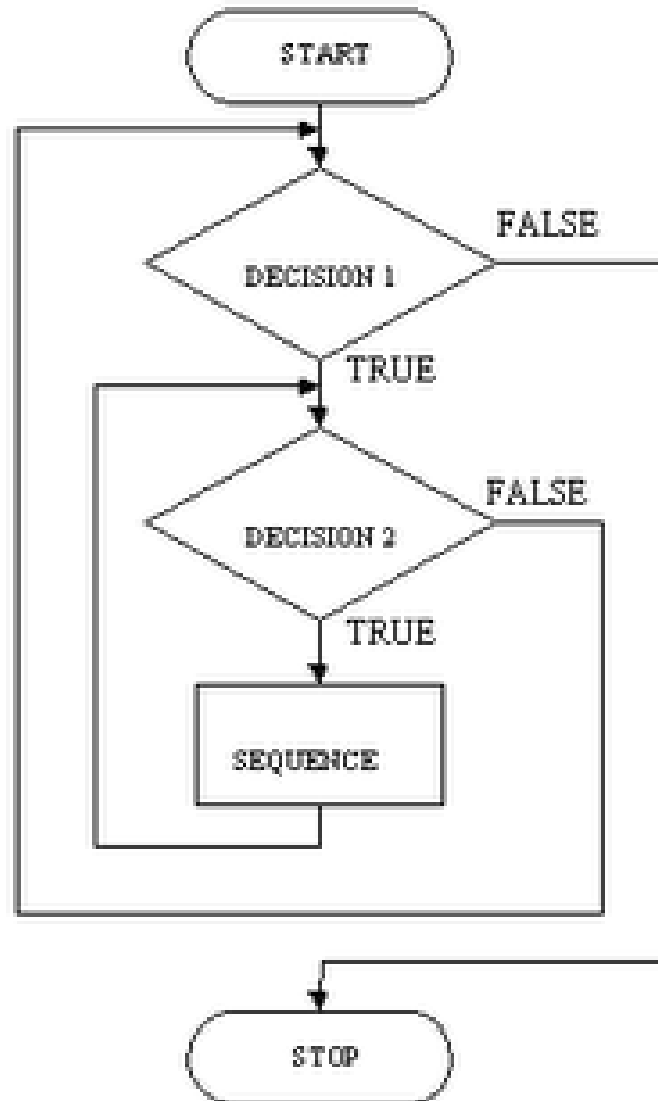
รับข้อมูล

เช็คว่าตัวเลขนั้นเป็นจำนวนเฉพาะหรือไม่ :  
ถ้า  $n$  หารจำนวนใด ๆ ลงตัวมากกว่า 2 จำนวน  
จะไม่ใช่จำนวนเฉพาะ

```
#include<stdio.h>
int main(){
    int n,i=1,count=0;
    scanf("%d",&n);
    while(i<=n){
        if(n%i==0){
            count++;
        }
        i++;
    }
    if(count==2){
        printf("%d is prime number. ",n);
    }else{
        printf("%d is not prime number. ",n);
    }
}
```

# การใช้ Loop ซ้อน Loop

เป็นการทำงานที่มีความซับซ้อนมากขึ้น โดยการนำ loop ซ้อนกันหลายๆชั้น มักจะใช้ในการประมวลผลข้อมูลในรูปแบบที่ซับซ้อน เช่น เมทริกซ์ เป็นต้น ซึ่งมีลักษณะการทำงานดังรูป



# มือใหม่ควรคิดแก้ปัญหาทีละส่วน

\* \* \* \* \*

\* N ตัว

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* NxN

\* - \* - \*

- \* - \* -

\* - \* - \*

- \* - \* -

\* - \* - \*

ใส่ \*สลับกับ -

**ตัวอย่าง** จงเขียนโปรแกรมรับจำนวนเต็ม  
1 ตัว เก็บไว้ในตัวแปร d จากนั้นให้พิมพ์  
รูปสี่เหลี่ยมขนาด dxd โดยใช้ \*

### Input-Output

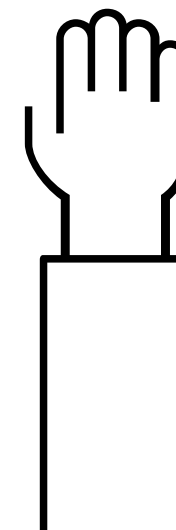
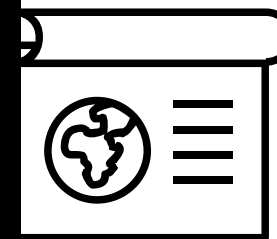
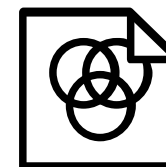
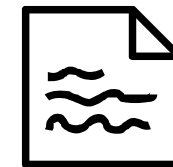
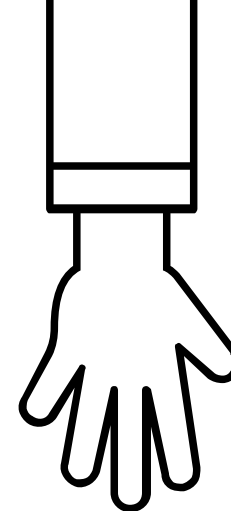
Enter a number : 3

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

### Input-Output

Enter a number : 6

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



i	j	1	2	3
1		*	*	*
2		*	*	*
3		*	*	*

```
#include<stdio.h>
int main() {
    for(int i=1;i<=3;i++){
        for(int j=1;j<=3;j++){
            printf("*");
        }
        printf("\n");
    }
}
```

\*

—

i \ j	1	2	3	4	5
1	i=1 j=1	i=1 j=2	i=1 j=3	i=1 j=4	i=1 j=5
2	i=2 j=1	i=2 j=2	i=2 j=3	i=2 j=4	i=2 j=5
3	i=3 j=1	i=3 j=2	i=3 j=3	i=3 j=4	i=3 j=5
4	i=4 j=1	i=4 j=2	i=4 j=3	i=4 j=4	i=4 j=5
5	i=5 j=1	i=5 j=2	i=5 j=3	i=5 j=4	i=5 j=5

สรุปความสัมพันธ์/รูปแบบ

$i+j$  = จำนวนคู่ : ดอกจันทร์  
 $i+j$  = จำนวนคี่ : ชีด



\* \_ \* \_ \*  
\_ \* \_ \* \_  
\* \_ \* \_ \*  
\_ \* \_ \* \_  
\* \_ \* \_ \*

```
#include<stdio.h>
int main() {
    int n=5;
    int i;
    for(i=1;i<=n;i++) {
        for(int j=1;j<=n;j++) {
            if((i+j)%2==0)
                printf("*") ;
            else printf("-") ;
        }

        printf("\n") ;
    }
}
```

**ตัวอย่าง** จงเขียนโปรแกรมรับจำนวนเต็ม  
1 ตัว เก็บไว้ในตัวแปร n จากนั้นให้พิมพ์รูป  
สามเหลี่ยมที่มีความสูง d บรรทัด โดยใช้ \*

### Input-Output

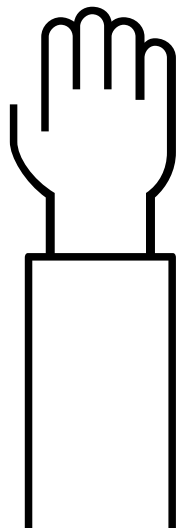
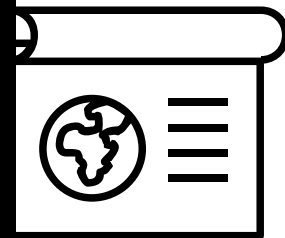
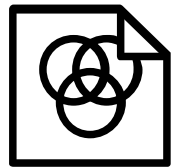
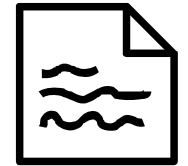
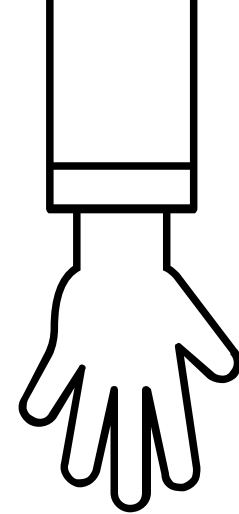
Enter a number : 3

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

### Input-Output

Enter a number : 5

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



i	j	1	2	3
1		*		
2		*	*	
3		*	*	*

```

#include<stdio.h>
int main() {
    for(int i=1;i<=3;i++) {
        for(int j=1;j<=i;j++) {
            printf("*");
        }
        printf("\n");
    }
}

```

**ตัวอย่าง** จงเขียนโปรแกรมรับจำนวนเต็ม  
1 ตัว เก็บไว้ในตัวแปร n จากนั้นให้พิมพ์รูป  
สามเหลี่ยมที่มีความสูง n บรรทัด โดยใช้ \*

### Input-Output

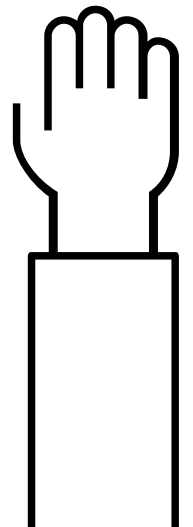
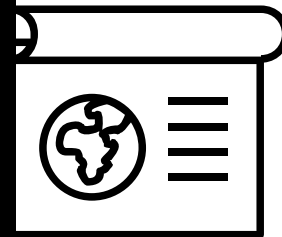
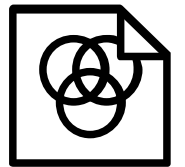
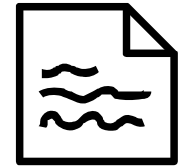
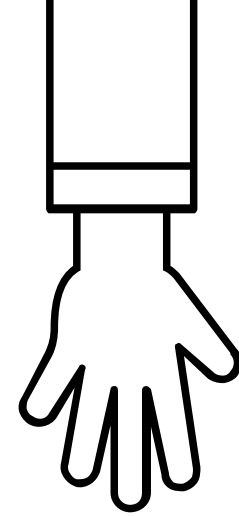
Enter a number : 3

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

### Input-Output

Enter a number : 5

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



i	j	1	2	3	4	5
1		i=1 j=1	i=1 j=2	i=1 j=3	i=1 j=4	i=1 j=5
2		i=2 j=1	i=2 j=2	i=2 j=3	i=2 j=4	i=2 j=5
3		i=3 j=1	i=3 j=2	i=3 j=3	i=3 j=4	i=3 j=5
4		i=4 j=1	i=4 j=2	i=4 j=3	i=4 j=4	i=4 j=5
5		i=5 j=1	i=5 j=2	i=5 j=3	i=5 j=4	i=5 j=5

i	j	1	2	3	4	5
1		i=1 j=1	i=1 j=2	i=1 j=3	i=1 j=4	i=1 j=5
2		i=2 j=1	i=2 j=2	i=2 j=3	i=2 j=4	i=2 j=5
3		i=3 j=1	i=3 j=2	i=3 j=3	i=3 j=4	i=3 j=5
4		i=4 j=1	i=4 j=2	i=4 j=3	i=4 j=4	i=4 j=5
5		i=5 j=1	i=5 j=2	i=5 j=3	i=5 j=4	i=5 j=5

```
#include<stdio.h>
int main() {
    for(int i=1;i<=5;i++){
        for(int j=1;j<=5;j++){
            if(i+j<=5)
                printf(" ");
            else printf("*");
        }
        printf("\n");
    }
}
```

**ตัวอย่าง** จงเขียนโปรแกรมรับจำนวนเต็ม  
1 ตัว เก็บไว้ในตัวแปร n จากนั้นให้พิมพ์รูป  
สามเหลี่ยมที่มีความสูง n บรรทัด โดยใช้ \*

### Input-Output

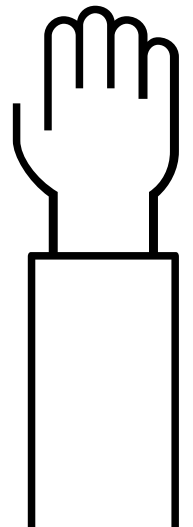
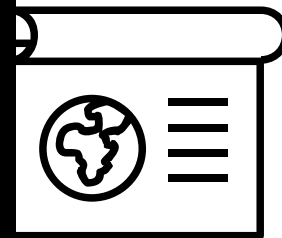
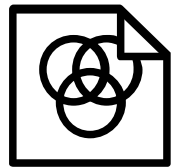
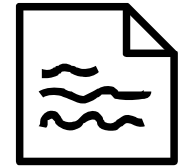
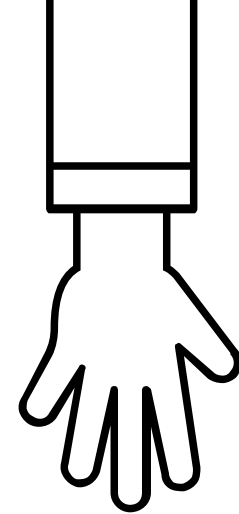
Enter a number : 3

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

### Input-Output

Enter a number : 5

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



row=5

i	j	1	2	3	4	5	6	7	8	9
1		i=1 j=1	i=1 j=2	i=1 j=3	i=1 j=4	*				
2		i=2 j=1	i=2 j=2	i=2 j=3	*	*	*			
3		i=3 j=1	i=3 j=2	*	*	*	*	*		
4		i=4 j=1	*	*	*	*	*	*	*	
5		*	*	*	*	*	*	*	*	*

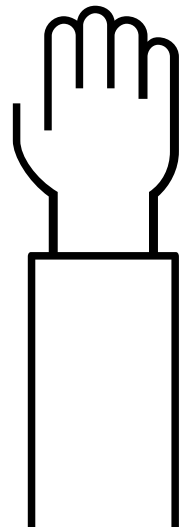
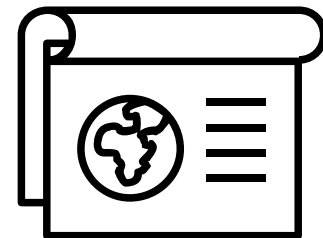
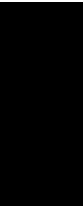
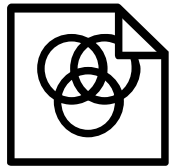
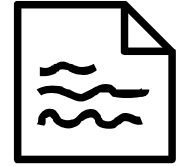
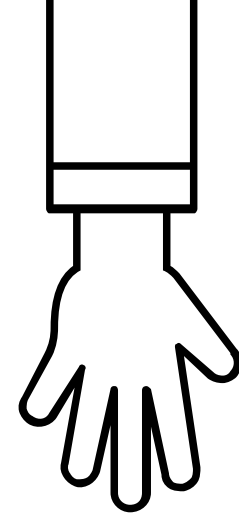
i	จำนวนspace		row-i	
1		4	5-1	
2		3	5-2	
3		2	5-3	
4		1	5-4	
5		0	5-5	

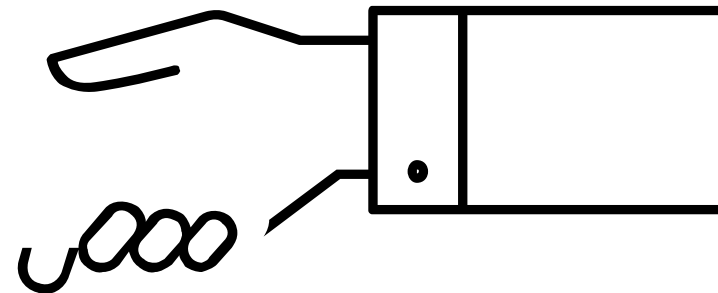
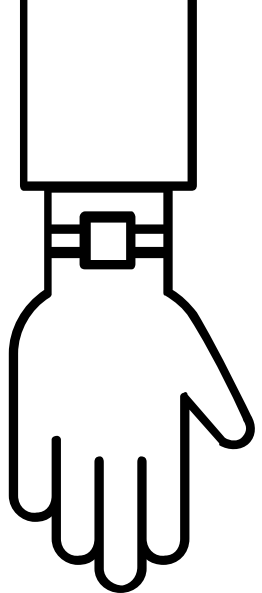
i	จำนวน *		2*i-1	
1		1	2*1-1	
2		3	2*2-1	
3		5	2*3-1	
4		7	2*4-1	
5		9	2*5-1	



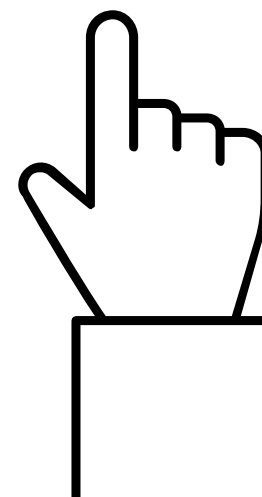
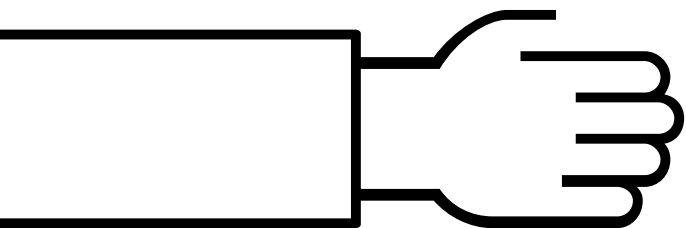
```
#include<stdio.h>
int main(){
    int row=5;
    int i;
    for(i=1;i<=row;i++){
        for(int j=1;j<=row-i;j++){
            printf(" ");

        }
        for(int j=1;j<=2*i-1;j++){
            printf("*");
        }
        printf("\n");
    }
}
```





# ตัวอย่างโปรแกรม ประยุกต์

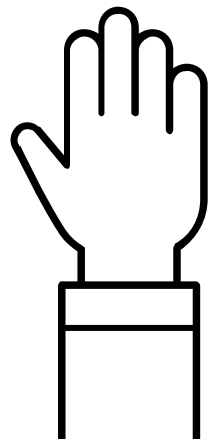
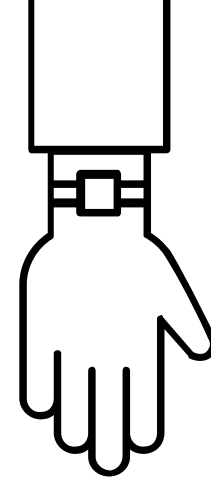


**ตัวอย่าง** จงเขียนโปรแกรมคำนวณระยะทางที่  
เวลาต่างๆ ตั้งแต่วินาทีที่ 2 ถึง 20  
ดังตัวอย่าง โดยกำหนดอินพุตของระบบคือ  
ความเร็วต้น (u) และความเร่ง (a)

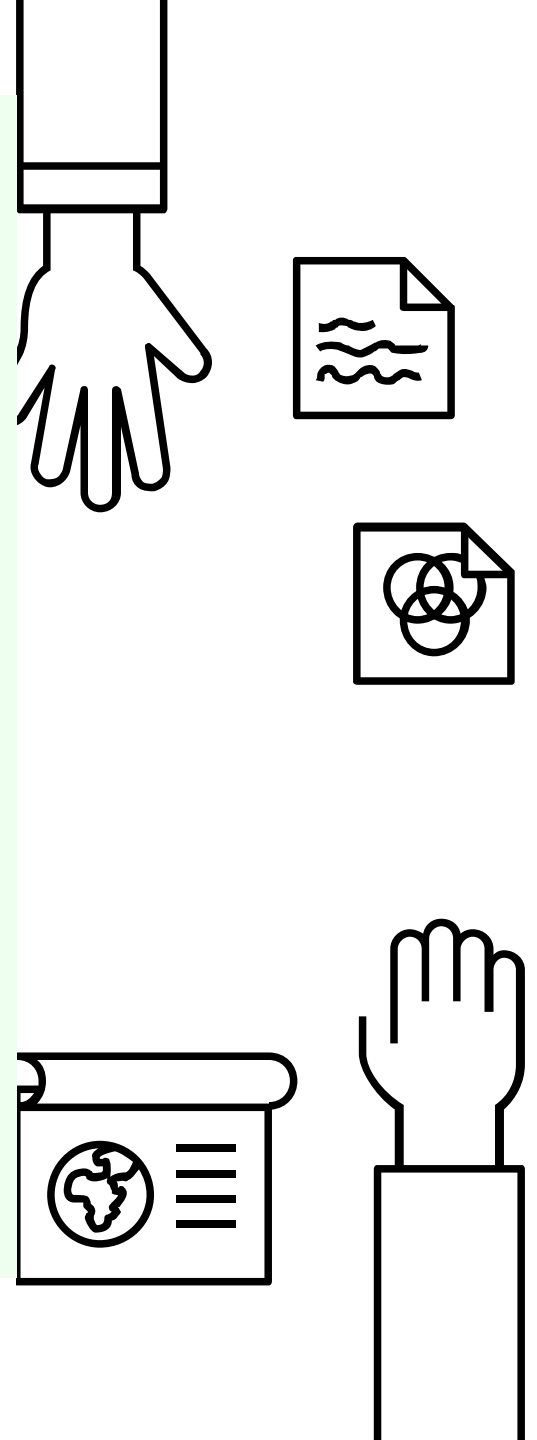
$$S = ut + \frac{1}{2}at^2$$

### Input-Output

```
Enter u : 4
Enter a : 2
S(2) = 12.00 m
S(4) = 32.00 m
S(6) = 60.00 m
S(8) = 96.00 m
S(10) = 140.00 m
S(12) = 192.00 m
S(14) = 252.00 m
S(16) = 320.00 m
S(18) = 396.00 m
S(20) = 480.00 m
```



```
#include <stdio.h>
int main()
{
    int u,a,t;
    float s;
    printf("Enter u : ");
    scanf("%d",&u);
    printf("Enter a : ");
    scanf("%d",&a);
    for(t=2;t<=20;t+=2){
        s=u*t+0.5*a*t*t;
        printf("S(%d) = %1.2f m\n",t,s);
    }
    return 0;
}
```



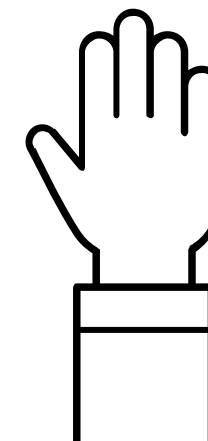
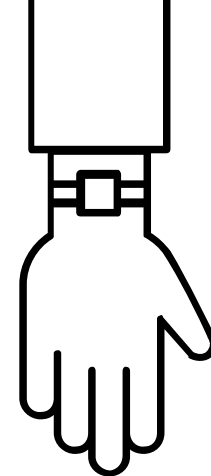
**ตัวอย่าง** จงเขียนโปรแกรมรับจำนวนเต็ม 1 ตัว แล้วแสดงลำดับฟีโบนัชชี (Fibonacci numbers) ตามจำนวนที่รับเข้ามา

$$x_n = x_{n-1} + x_{n-2} \quad \text{เมื่อ} \quad x_0=0, x_1=1$$

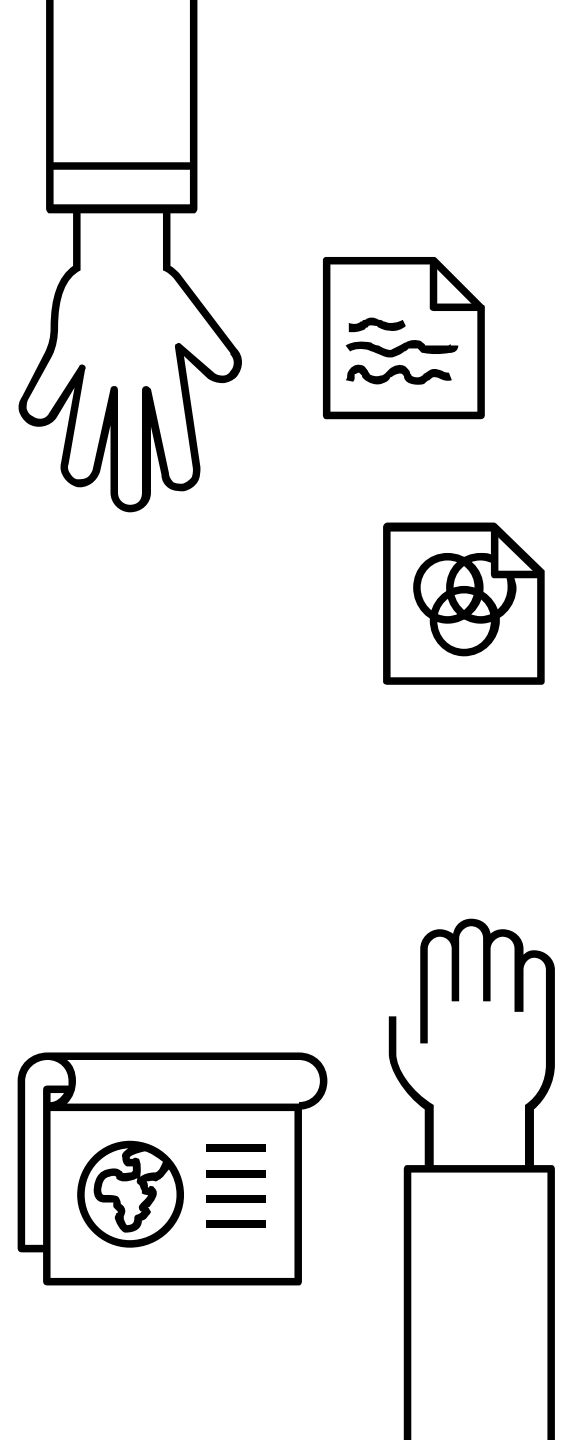
### Input-Output

Enter number : 15

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377



```
#include <stdio.h>
int main()
{
    int num,x0=0,x1=1,x2,i;
    printf("Enter number : ");
    scanf("%d",&num);
    if(num>=1){
        printf("%d ",x0);
    }
    if(num>=2){
        printf("%d ",x1);
    }
    for(i=3;i<=num;i++){
        x2=x1+x0;
        printf("%d ",x2);
        x0=x1;
        x1=x2;
    }
    return 0;
}
```





โปรแกรมเมอร์ไทย Thai programmer

12 มีนาคม เวลา 17:44 น. · 🌐



จุดแข็ง = เขียนโค้ดคุยกับคอมได้  
จุดอ่อน = คุยกับคนไม่รู้



2.6 พัน

ความคิดเห็น 78 รายการ แชร์ 695 ครั้ง

แบบฝึกหัด





## แบบฝึกหัดวันนี้^^

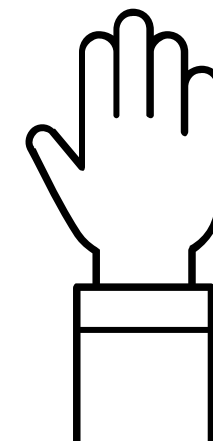
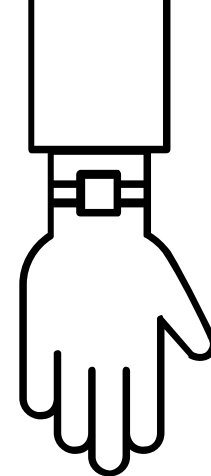
(ส่งใน Google Classroom)

4.1 โปรแกรมรับจำนวนเต็มบวกที่มีค่าตั้งแต่ 2  
1 จำนวน แล้วคำนวณหาผลคูณตัวเลขตั้งแต่ 2  
ถึง จำนวนที่รับมา แสดงผลดังตัวอย่าง

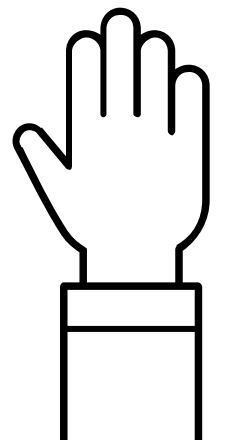
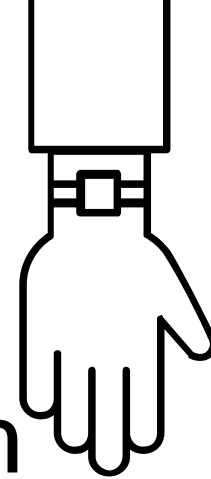
### Input-Output

Enter a positive number : 7

2 x 3 x 4 x 5 x 6 x 7 = 5040



4.2 รับจำนวนข้อมูล N ค่า  
แล้วกรอกข้อมูลตามจำนวนที่รับ โดยแต่ละค่า  
อยู่ระหว่าง 0-100 แล้วให้คอมพิวเตอร์แจ้งว่า  
ค่า MAX และ MIN คือค่าใด พร้อมบอก  
ผลรวมและค่าเฉลี่ย เมื่อป้อนตัวเลขไม่ตรงกับ  
ช่วงระหว่าง 0-100 ให้แจ้ง Invalid input  
แล้วให้โปรแกรมหยุดทำงาน



4.3. เขียนโปรแกรมรับเลขจำนวนเต็ม 1 จำนวน  
แล้วแสดงผลลัพท์ย้อนกลับ  
ถ้าค่าน้อยกว่า 0 ให้แสดง invalid input

Input :	38	5673	980123	47629145
Output :	83	3765	321089	54192674

สำหรับการ input 1000

Output สำหรับมือใหม่ : 1

แล้วลอง พัฒนาได้ให้เป็น : 0001

## 4.4 เขียนโปรแกรมรับเลขจำนวนเต็มบวก แล้ว แสดงผลดังรูป ถ้าค่าน้อยกว่า 1 ให้แสดง invalid input

Input :	1	3	5	6
Output :	1	1 2 4 3 6 9	1 2 4 3 6 9 4 8 12 16 5 10 15 20 25	1 2 4 3 6 9 4 8 12 16 5 10 15 20 25 6 12 18 24 30 36

4.5 ให้รับค่าจำนวนเต็มบวก N โดยค่า N มีค่าไม่เกิน 26 แล้วให้แสดงผลเป็นตัวอักษรภาษาอังกฤษ N บรรทัด โดยเรียงลำดับอักษรในแต่ละบรรทัด  
ถ้าค่าน้อยกว่า 0 ให้แสดง invalid input

### Input-Output

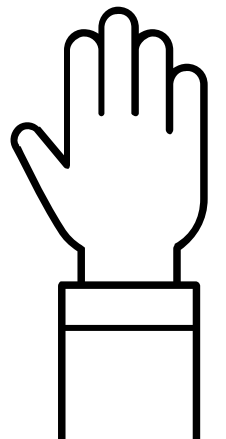
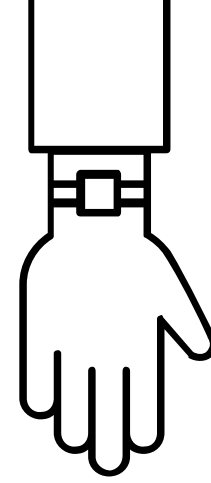
Enter a number : 4

A

AB

ABC

ABCD



4.6 ให้รับค่าจำนวนเต็มบวก N โดยค่า N มีค่าน้อยกว่าหรือเท่ากับ 200 แล้วให้แสดงผล ดังต่อไปนี้ (โดยหากค่าที่รับเข้ามามีค่ามากกว่าจำนวนตัวอักษรภาษาอังกฤษให้วนค่า A ใหม่) ถ้าค่าน้อยกว่า 0 ให้แสดง invalid input

### Input-Output

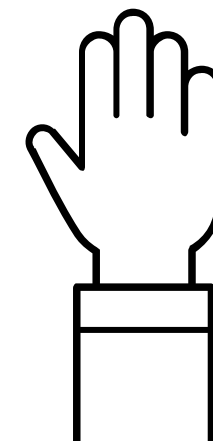
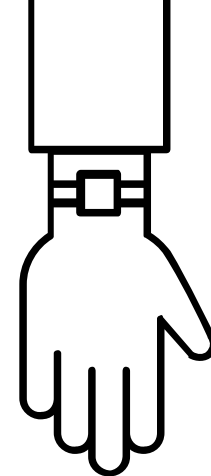
Enter a number : 4

A

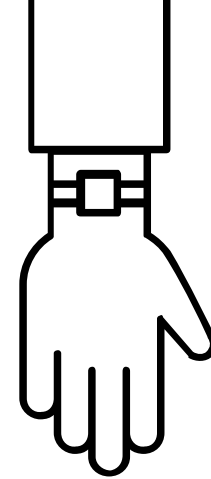
BC

DEF

GHIJ



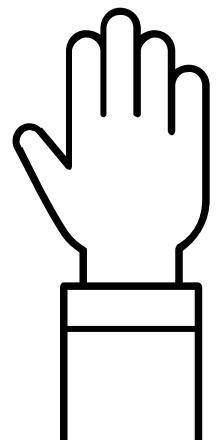
## 4.7 เขียนโปรแกรมรับเลขจำนวนเต็มบวก แล้วแสดงผลดังรูป



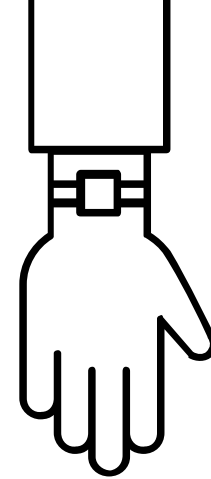
Input-  
Enter : 1  
0

Input-  
Enter : 3  
0  
101  
21012

Input-Output  
Enter : 5  
0  
101  
21012  
3210123  
432101234



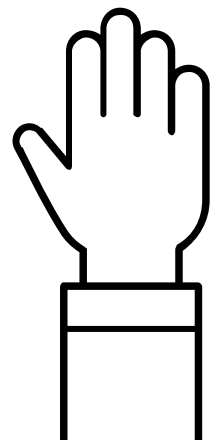
## 4.8 เขียนโปรแกรมรับเลขจำนวนเต็มบวก แล้วแสดงผลดังรูป



Input-  
Enter : 1  
1

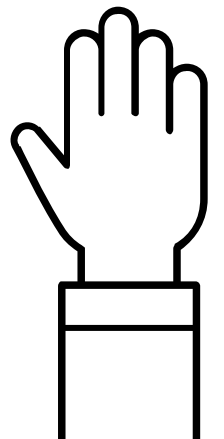
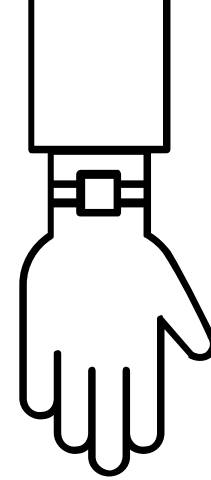
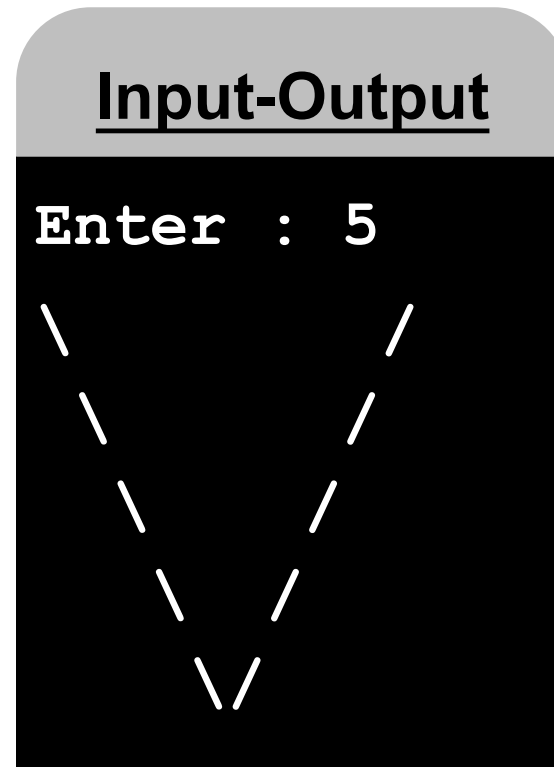
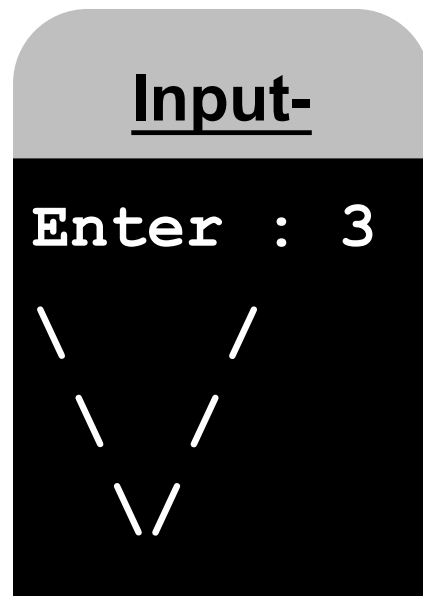
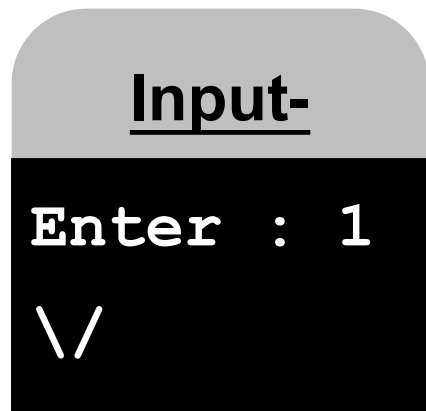
Input-  
Enter : 3  
1  
121  
12321

Input-Output  
Enter : 5  
1  
121  
12321  
1234321  
123454321





## 4.9 เขียนโปรแกรมรับเลขจำนวนเต็ม บวก เป็นความสูงของหินงอกหินย้อย แล้วแสดงผลดังรูป



4.10. เขียนโปรแกรมรับเลขจำนวนเต็มบวก แล้วแสดงผลดังรูป (โดยหากค่าที่รับเข้ามามีค่ามากกว่าจำนวนตัวอักษรภาษาอังกฤษให้วนค่า A ใหม่)

Input :	1	3	5	6
Output :	A	A CB FED	A CB FED JIHG ONMLK	A CB FED JIHG ONMLK UTSRQP