

### 1.1 Output Analysis

Num 1	=	50
Num 2	=	50
Sum	=	100

### 1.2 Input Analysis

รับค่าจำนวน 2 ค่า

Num 1, Num 2

### 1.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลข เก็บในตัวแปร Num 1
3. รับค่าตัวเลข เก็บในตัวแปร Num 2
4. คำนวณหาผลรวมตัวเลข 2 จำนวน  
เก็บผลลัพธ์ไว้ในตัวแปร Sum
5. แสดงผลค่าในตัวแปร Sum บนจอภาพ
6. จบโปรแกรม

### 1.4 Variable Defined

Num 1 เป็นชนิดจำนวนเต็ม (Int)

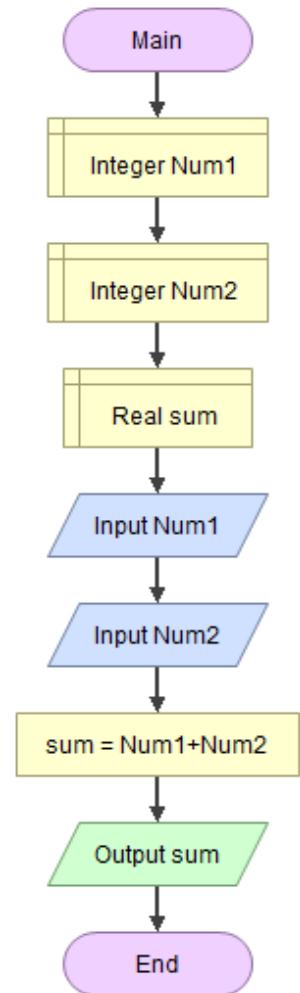
เก็บค่าตัวเลขที่ 1

Num 2 เป็นชนิดจำนวนเต็ม (Int)

เก็บค่าตัวเลขที่ 2

Sum เป็นชนิดจำนวนเต็ม (Int)

เก็บผลลัพธ์การบวกเลข 2 จำนวน



## 2.1 Output Analysis

Select function Tor 5:5

Input Length : 10

Area Square : 20

## 2.2 Input Analysis

รับค่า เลือกฟังก์ชัน (Triangle, Square)

ฟังก์ชัน Triangle หาพื้นที่สามเหลี่ยม

รับค่า 2 จำนวน (Length, High)

ฟังก์ชัน Square หาพื้นที่สี่เหลี่ยมจัตุรัส

รับค่า 1 จำนวน (Length)

## 2.3 Process Analysis

1. เริ่มต้น

2. เลือกฟังก์ชันที่ต้องการหา

(1) Triangle () (พื้นที่สามเหลี่ยม)

(2) Square () (พื้นที่สี่เหลี่ยมจัตุรัส)

3. ถ้าหากเลือกฟังก์ชัน Triangle ()

จะให้ใส่ค่าความสูง,ความยาว เก็บในตัวแปร Length, High

ถ้าหากเลือกฟังก์ชัน Square () จะให้ใส่ค่าความยาว เก็บในตัวแปร Length

4. คำนวณหาค่าพื้นที่ตามฟังก์ชันที่เลือก

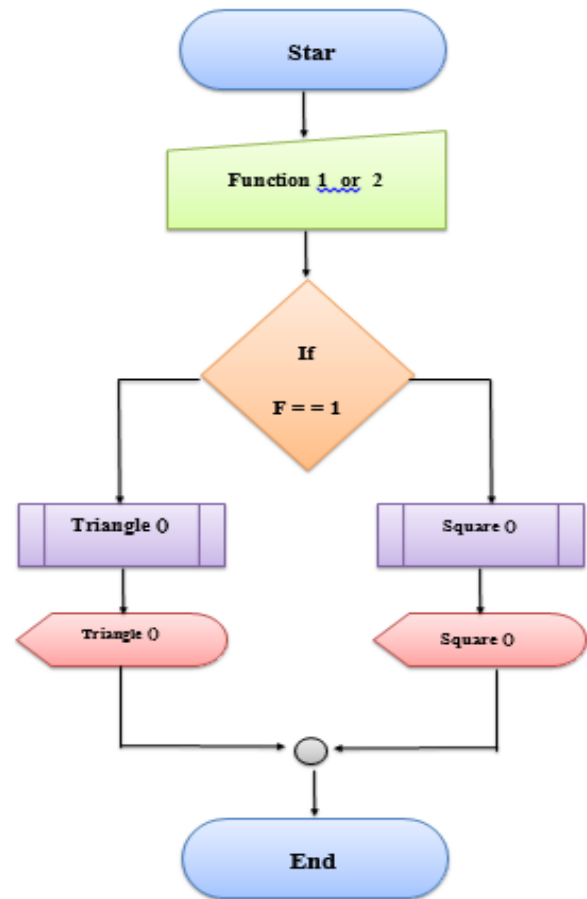
5. แสดงผลของฟังก์ชันที่คำนวณได้ออกจอภาพ

6. จบโปรแกรม

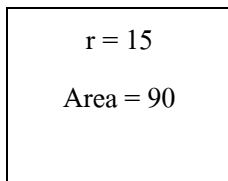
## 2.4 Variable Defined

Length เป็นชนิดจำนวนเต็ม (Int) รับค่าความยาว

High เป็นชนิดจำนวนเต็ม (Int) รับค่าความสูง



### 3.1 Output Analysis



### 3.2 Input Analysis

รับตัวเลขเก็บไว้ในตัวแปร r

1 จำนวน

### 3.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลข เก็บไว้ในตัวแปร r
3. คำนวณค่าโดยใช้สูตร
4. พื้นที่สี่เหลี่ยม - พื้นที่วงกลม  
 $(2r)^2 - \pi r^2$   
เก็บไว้ในตัวแปร Area
5. แสดงค่าในตัวแปร Area ออกสู่จอภาพ
6. จบโปรแกรม

### 3.4 Variable Defined

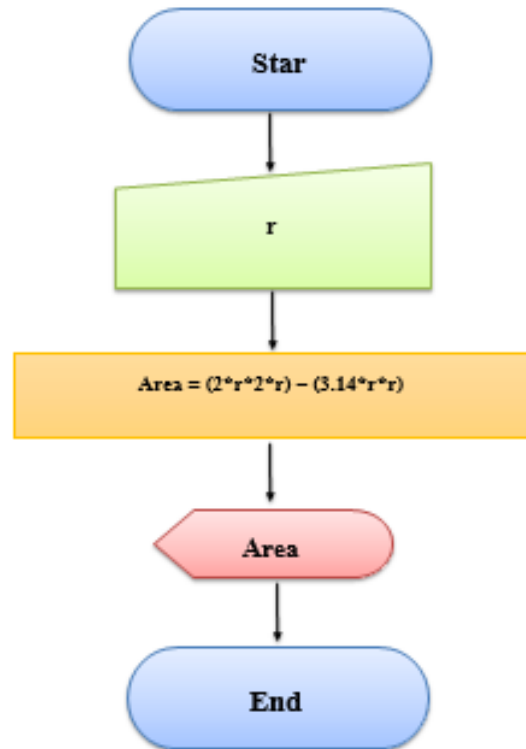
r เป็นชนิดทศนิยม (float)

เก็บค่าตัวเลข

Area เป็นชนิดทศนิยม (float)

เก็บผลลัพธ์ การหาพื้นที่สี่เหลี่ยม

(ในส่วนของแรงงา)



#### 4.1 Output Analysis

Num = 50  
it is even number

Num = 51  
it is odd  
number

#### 4.2 Input Analysis

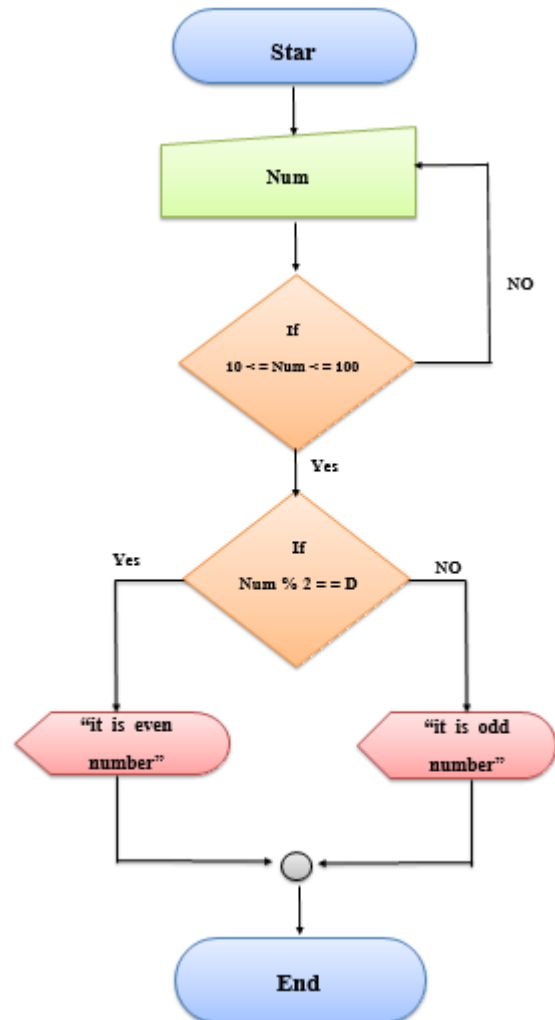
รับค่า 1 จำนวน (Num)

#### 4.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลข เก็บไว้ในตัวแปร Num
3. ตรวจสอบค่า Num  
ถ้าอยู่ในช่วง 10 – 100 ดำเนินการต่อ  
ถ้าไม่ใช่ให้กลับไปรับค่า Num ใหม่
4. ตรวจสอบเงื่อนไขถ้า  $\text{Num} \bmod 2$  เท่ากับ 0  
แสดงค่า it is even number สู่อุปกรณ์  
ถ้า  $\text{Num} \bmod 2$  ไม่เท่ากับ 0  
แสดงค่า it is odd number สู่อุปกรณ์
5. จบโปรแกรม

#### 4.4 Variable Defined

Num เป็นตัวแปรชนิดจำนวนเต็ม (Int)  
เก็บค่าตัวเลข



### 5.1 Output Analysis

Num 1	=	35
Num 2	=	15
Num 3	=	40
Sum	=	90
Two - Digit		

### 5.2 Input Analysis

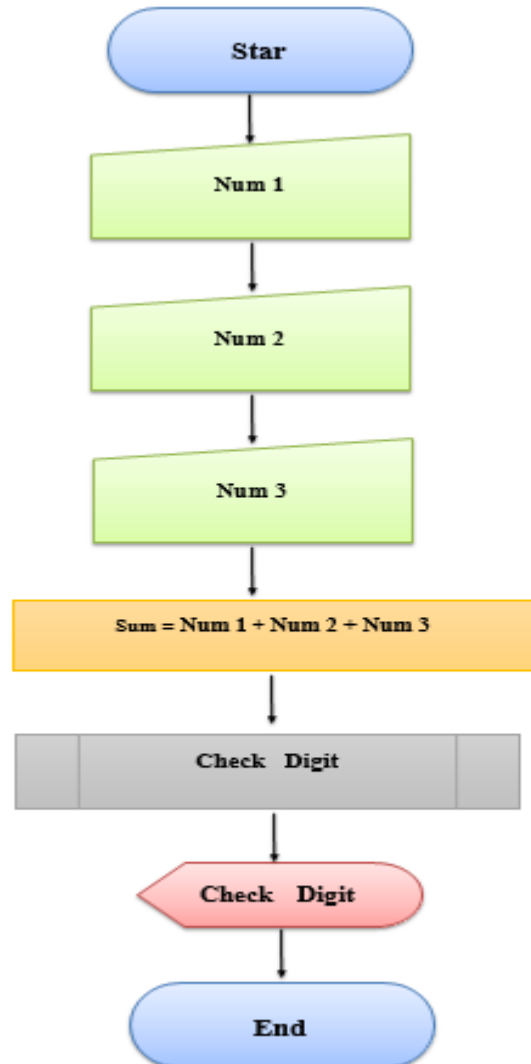
รับค่าตัวเลข 3 ค่า (Num 1, Num 2, Num 3)

### 5.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลข เก็บไว้ในตัวแปร Num 1
3. รับค่าตัวเลข เก็บไว้ในตัวแปร Num 2
4. รับค่าตัวเลข เก็บไว้ในตัวแปร Num 3
5. คำนวณหาผลรวมตัวเลข 3 จำนวน  
เก็บไว้ในตัวแปร Sum
6. แสดงผลค่าที่เก็บไว้ในตัวแปร Sum  
บนจอภาพ
7. ตรวจสอบค่า Sum ค่าผลลัพธ์เป็นเลขกี่หลัก  
ถ้า 1 หลัก แสดง One - Digit นอกจากนี้ แสดง Over - Three - Digit  
2 หลัก แสดง Two - Digit  
3 หลัก แสดง Three - Digit
8. จบโปรแกรม

### 5.4 Variable Defined

- Num 1 เป็นชนิดจำนวนเต็ม (Int) เก็บค่าตัวเลขที่ 1
- Num 2 เป็นชนิดจำนวนเต็ม (Int) เก็บค่าตัวเลขที่ 2
- Num 3 เป็นชนิดจำนวนเต็ม (Int) เก็บค่าตัวเลขที่ 3
- Sum เป็นชนิดจำนวนเต็ม (Int) เก็บผลลัพธ์การบวกเลข 3 จำนวน



### 6.1 Output Analysis

input score = 88  
Grade B+

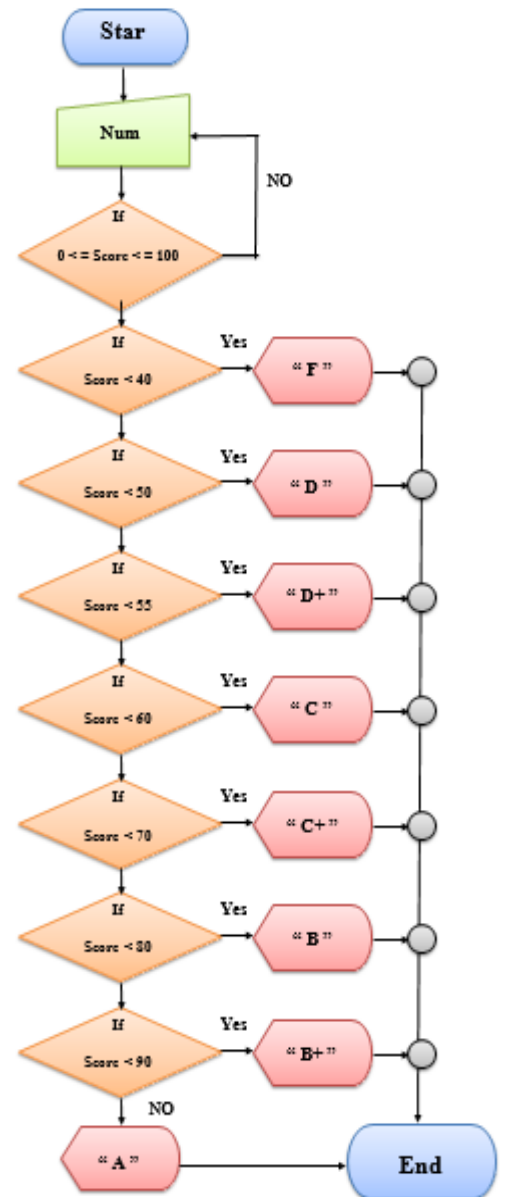
input score = 120  
input score = 95  
Grade A

### 6.2 Input Analysis

รับค่าตัวเลข 1 จำนวน (score)

### 6.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลข เก็บไว้ในตัวแปร score
3. ตรวจสอบเงื่อนไขว่าค่า score อยู่ในช่วง 0 – 100 หรือไม่  
ถ้าใช่ดำเนินการต่อ ถ้าไม่ใช่ กลับไปรับค่าตัวเลขใหม่  
แล้วเก็บไว้ในตัวแปร score
4. ตรวจสอบถ้าค่า score น้อยกว่า 40 ให้แสดงผล F  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
5. ตรวจสอบถ้าค่า score น้อยกว่า 50 ให้แสดงผล D  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
6. ตรวจสอบถ้าค่า score น้อยกว่า 55 ให้แสดงผล D+  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
7. ตรวจสอบถ้าค่า score น้อยกว่า 60 ให้แสดงผล C  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
8. ตรวจสอบถ้าค่า score น้อยกว่า 70 ให้แสดงผล C+  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
9. ตรวจสอบถ้าค่า score น้อยกว่า 80 ให้แสดงผล B  
และจบการทำงาน ถ้าไม่ใช่ดำเนินการต่อที่เงื่อนไขถัดไป
10. ตรวจสอบถ้าค่า score น้อยกว่า 90 ให้แสดงผล B+  
และจบการทำงาน ถ้าไม่ตรงเงื่อนไข ให้แสดงผล A และจบการทำงาน



### 6.4 Variable Defined

score เป็นชนิดจำนวนเต็ม (Int)

เก็บตัวเลขคะแนน

### 7.1 Output Analysis

Enter a – z or A – Z : X  
X  
Y  
Z

Enter a – z or A – Z : C  
C  
B  
A

### 7.2 Input Analysis

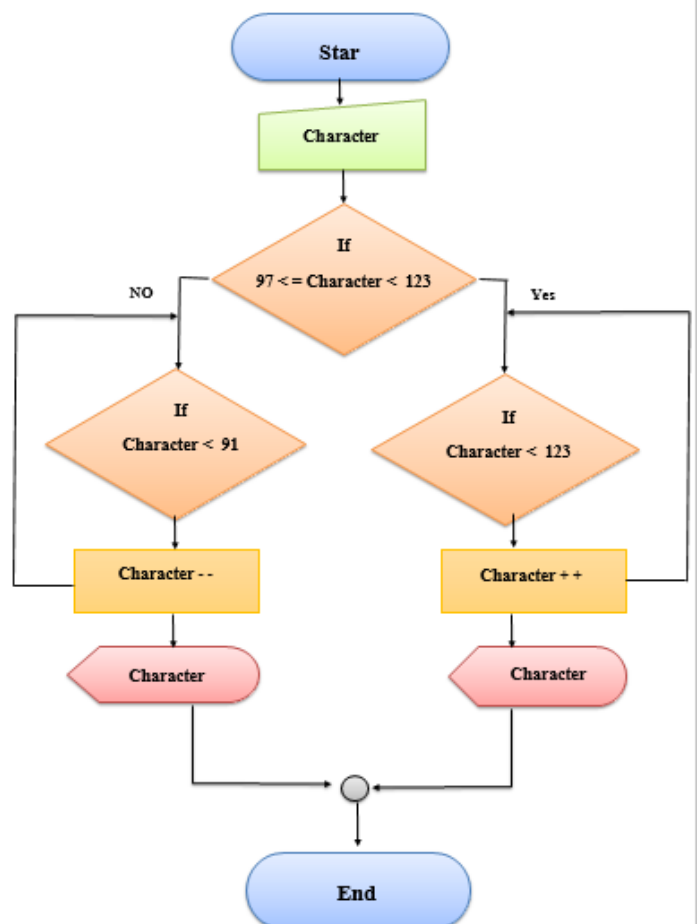
รับค่าตัวอักษร 1 ตัว (Char 1)

### 7.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวอักษรเก็บในตัวแปร Char 1
3. ตรวจสอบตัวอักษรว่าเป็น a – z หรือ A – Z หรือไม่ ถ้าไม่ใช่ให้กลับไปรับข้อมูลใหม่
4. ตรวจสอบตัวอักษรเป็นตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็ก
5. ถ้าเป็นอักษรพิมพ์เล็ก จะไล่จากตัวอักษรที่รับค่ามาแสดงไปถึง Z  
ถ้าเป็นอักษรพิมพ์ใหญ่ จะแสดงจากตัวอักษรตัวนั้นไปย้อนกลับไปถึง A  
(ใช้ For หรือ While ในการวน Loop ในโปรแกรม)
6. จบโปรแกรม

### 7.4 Variable Defined

Char 1 เป็นชนิดอักขระรับค่าตัวอักษร  
(Char)



### 8.1 Output Analysis

Num = 50  
it is even number

Num 33  
it is odd number

### 8.2 Input Analysis

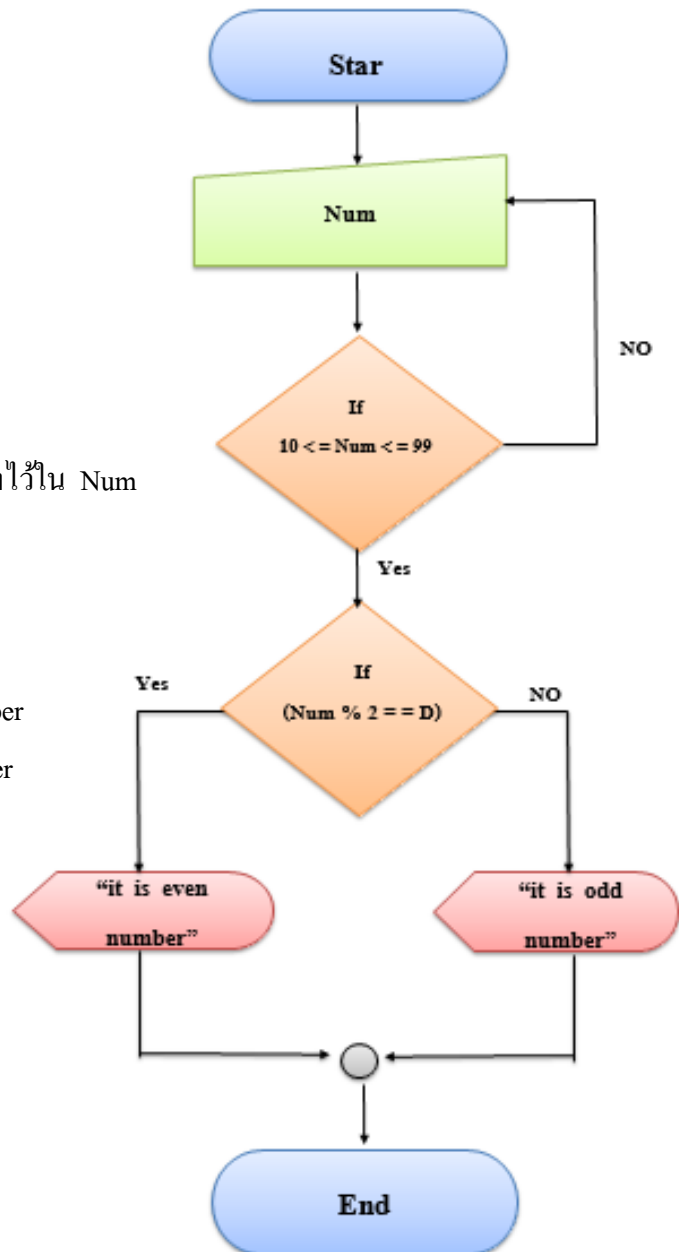
รับค่า 1 จำนวน (Num)

### 8.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลขเก็บในตัวแปร Num
3. ตรวจสอบค่า Num  
อยู่ในช่วง 10 – 100 ดำเนินการต่อ  
ถ้าไม่กลับไปรับค่าตัวเลขใหม่ แล้วเก็บค่าไว้ใน Num
4. ส่งค่าไปที่ฟังก์ชัน even\_or\_odd()
5. ประมวลผลฟังก์ชัน even\_or\_odd()
6. จากการประมวลผล  
ถ้าเป็นจำนวนคู่ แสดง it is even number  
ถ้าเป็นจำนวนคี่ แสดง it is odd number
7. จบการทำงาน

### 8.4 Variable Defined

Num เป็นชนิดจำนวนเต็ม (Int)  
เป็นค่าตัวเลข





### 9.1 Output Analysis

Num 1	=	60
Num 2	=	30
Sum	=	90

### 9.2 Input Analysis

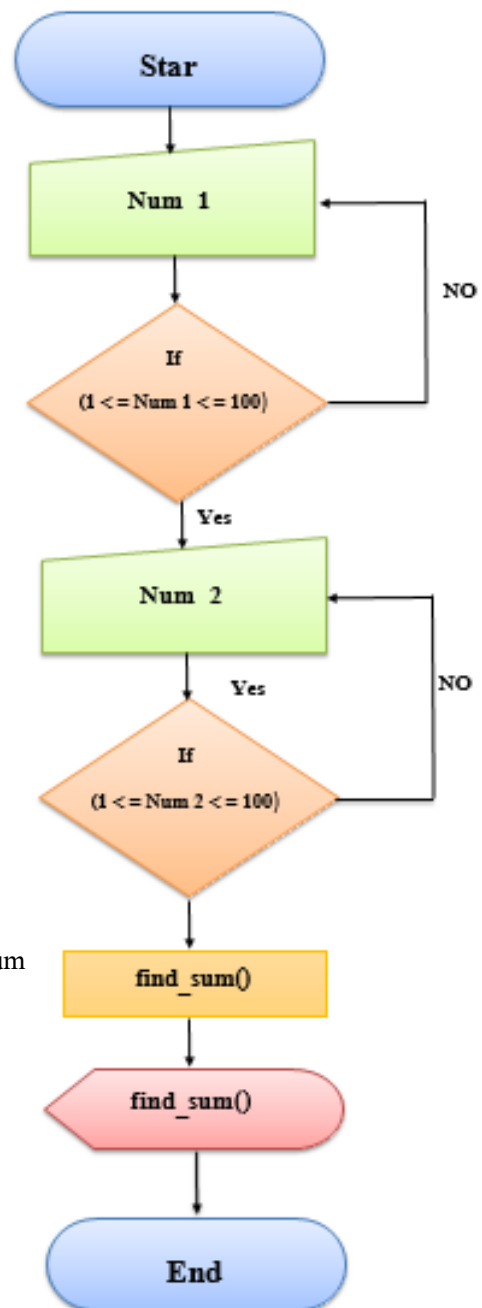
รับค่าตัวเลข (Num 1, Num 2)

### 9.3 Process Analysis

- เริ่มต้น
- รับค่าตัวเลขเก็บในตัวแปร Num 1
- ตรวจสอบตัวเลขค่า Num 1  
ว่าอยู่ในช่วง 0 – 100 หรือป่าว ถ้าใช่ดำเนินการต่อ  
ถ้าไม่กลับไปรับค่าตัวเลขใหม่
- รับค่าตัวเลขเก็บไว้ในตัวแปร Num 2
- ตรวจสอบตัวเลขค่า Num 2  
ว่าอยู่ในช่วง 0 – 100 ถ้าใช่ดำเนินการต่อ  
ถ้าไม่กลับไปรับค่าตัวเลขใหม่
- ส่งค่า Num 1, Num 2 ไปที่ฟังก์ชัน find\_sum()
- ประมวลผลรวมตัวเลขทั้ง 2 จำนวน เก็บไว้ในตัวแปร Sum
- แสดงค่าใน Sum
- จบการทำงาน

### 9.4 Variable Defined

- Num 1 เป็นชนิดจำนวนเต็ม (Int)  
เก็บค่าตัวเลขที่ 1
- Num 2 เป็นชนิดจำนวนเต็ม (Int)  
เก็บค่าตัวเลขที่ 2
- Sum เป็นชนิดจำนวนเต็ม  
เก็บค่าผลรวม Num 1, Num 2



### 10.1 Output Analysis

Num 1	=	60
Num 2	=	70
Average	=	65

### 10.2 Input Analysis

รับค่า 2 จำนวน (Num 1, Num 2)

### 10.3 Process Analysis

1. เริ่มต้น
2. รับค่าตัวเลขเก็บในตัวแปร Num 1
3. ตรวจสอบตัวเลขค่า Num 1  
ว่าอยู่ในช่วง 0 – 100 หรือไม่ ถ้าใช่ดำเนินการต่อ  
ถ้าไม่กลับไปรับค่าตัวเลขใหม่
4. รับค่าตัวเลขเก็บไว้ในตัวแปร Num 2
5. ตรวจสอบตัวเลขค่า Num 2  
ว่าอยู่ในช่วง 0 – 100 หรือไม่ ถ้าใช่ดำเนินการต่อ  
ถ้าไม่กลับไปรับค่าตัวเลขใหม่
6. ส่งค่า Num 1, Num 2 ไปที่ฟังก์ชัน find\_averge()  
(ฟังก์ชันหาค่าเฉลี่ย ของเลขทั้ง 2 จำนวน)
7. แสดงค่าเฉลี่ยของเลข 2 จำนวน (Num 1, Num 2)
8. จบการทำงาน

### 10.4 Variable Defined

- Num 1 เป็นชนิดจำนวนเต็ม (Int)  
เก็บค่าตัวเลขที่ 1
- Num 2 เป็นชนิดจำนวนเต็ม (Int)  
เก็บค่าตัวเลขที่ 2
- Average เป็นชนิดจำนวนเต็ม (Int)  
เก็บค่าเฉลี่ยตัวเลขทั้ง 2 ค่า

