CS F111: Computer Programming

(Second Semester 2021-22)

Lect 5: Flowchart & Algorithm

Dr. Nikumani Choudhury
Asst. Prof., Dept. of Computer Sc. & Information Systems nikumani@hyderabad.bits-pilani.ac.in

BITS Pilani
Hyderabad Campus

Algorithms, Flowcharts & Pseudocodes

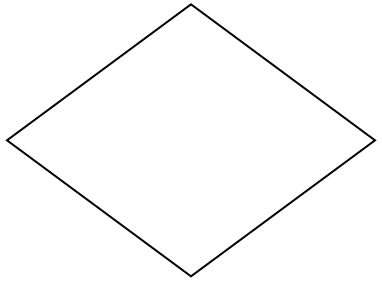
Algorithm: Stepwise solution to the given problem

- **Each step should be precise & unambiguous**
- **Each step should terminate**

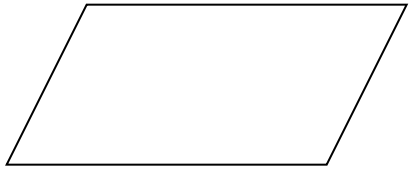
Flowchart: pictorial representation of algorithm

Flow Chart Symbols

Computation



Decision



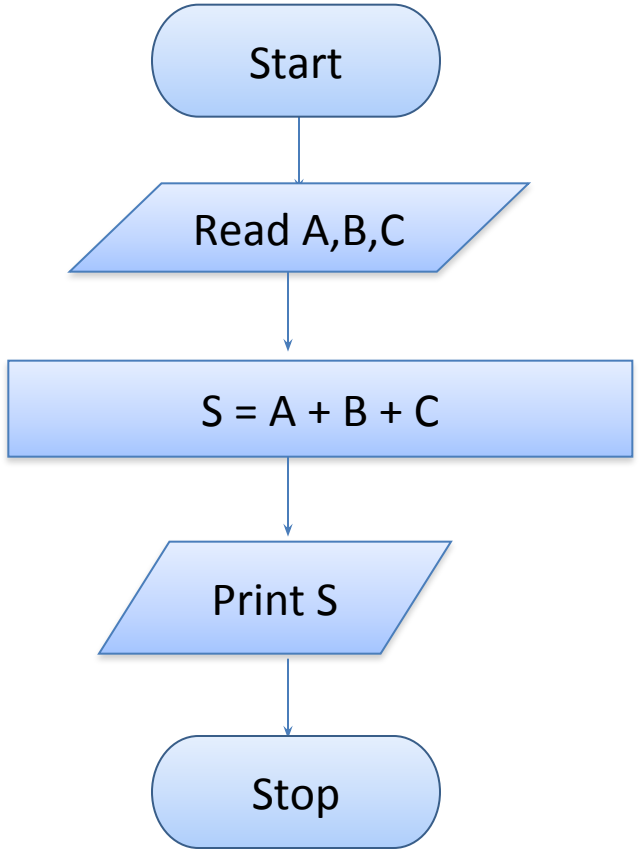
Input, Output

Flow Line

Termination
(Start, Stop)

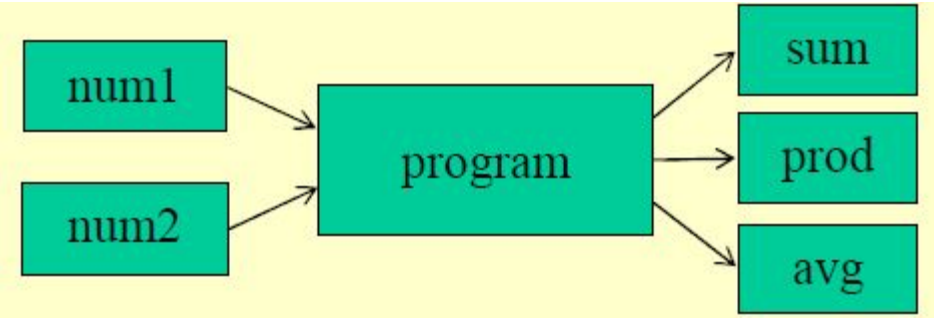
Example 1: *Adding three numbers*

1. START
2. Read A, B, C
3. S = A + B + C
4. Print S
5. STOP



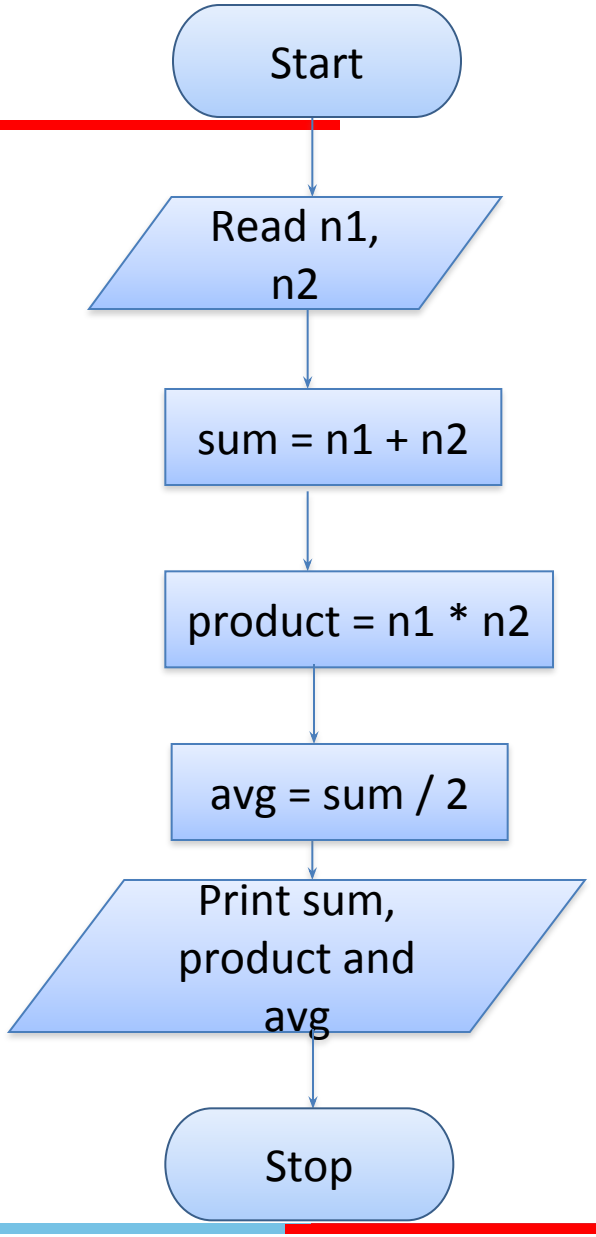
Ex2 - Compute the sum, product and average of 2 numbers

- **Problem: Design an algorithm to find the sum, product and average of 2 numbers**
- **Input**
  - **2 numbers**
- **Output**
  - **Sum, product and average of the numbers**



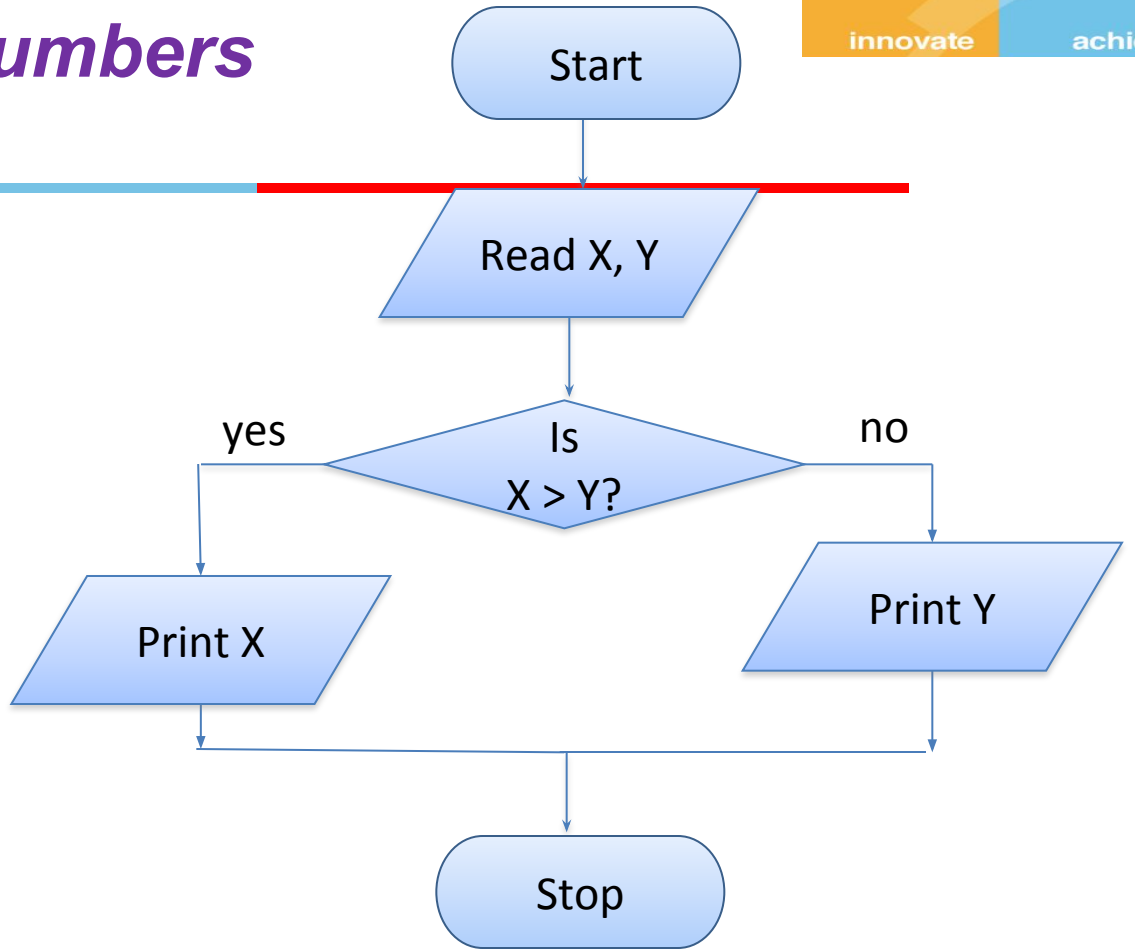
Example of Compute the sum, product and average of 2 numbers

1. START
2. Read n1,n2
3. sum = n1 + n2
4. product = n1 * n2;
5. average = sum / 2;
6. Print sum, product and average
7. STOP



Example 4: *Larger of two numbers*

1. START
2. Read X,Y
3. if X > Y
   3.1 Print X
4. else
   4.1 Print Y
5. STOP

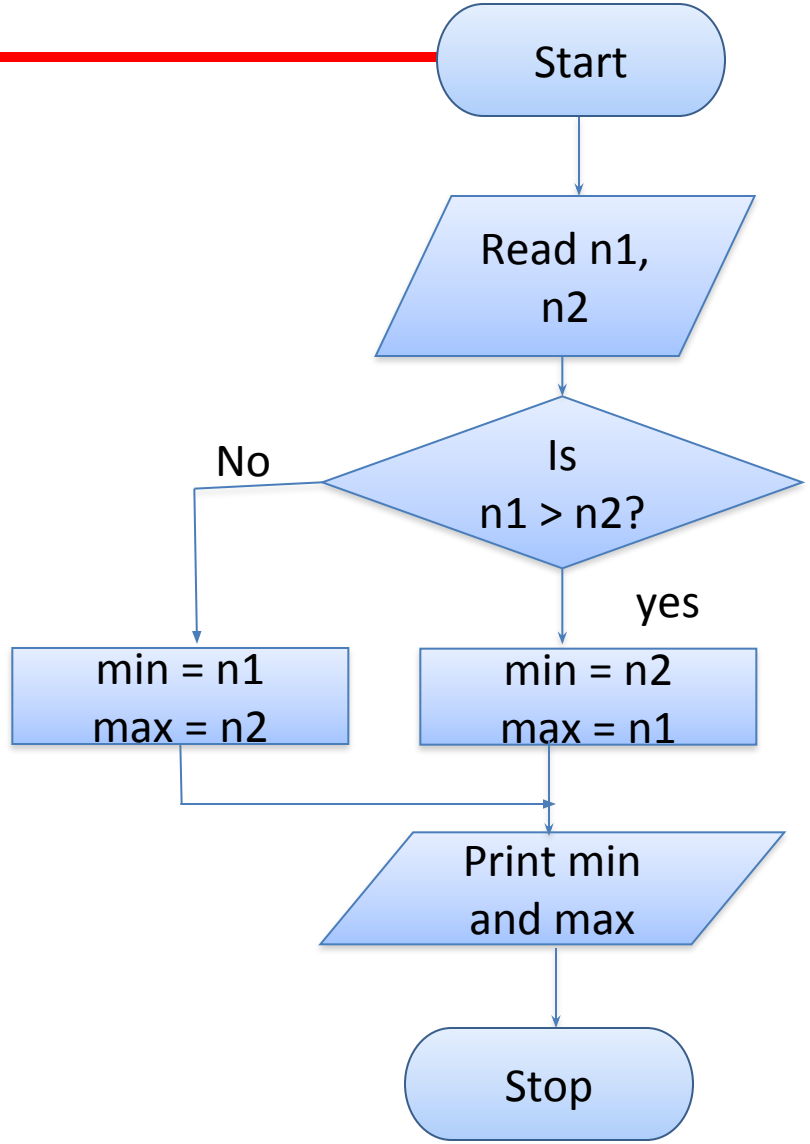


Ex 5 - Compute the minimum and maximum of 2 numbers

- **Problem: Design an algorithm to find the minimum and maximum of 2 numbers**
- **Input**
  - **2 numbers**
- **Output**
  - **Minimum and maximum of the numbers**

Ex 5 - Compute the minimum and maximum of 2 numbers

1. START
2. Read n1, n2
3. if n1 < n2
   - 3.1 min = n1
   - 3.2 max = n2
4. else
   - 4.1 min = n2
   - 4.2 max = n1
5. Print min and max
6. STOP



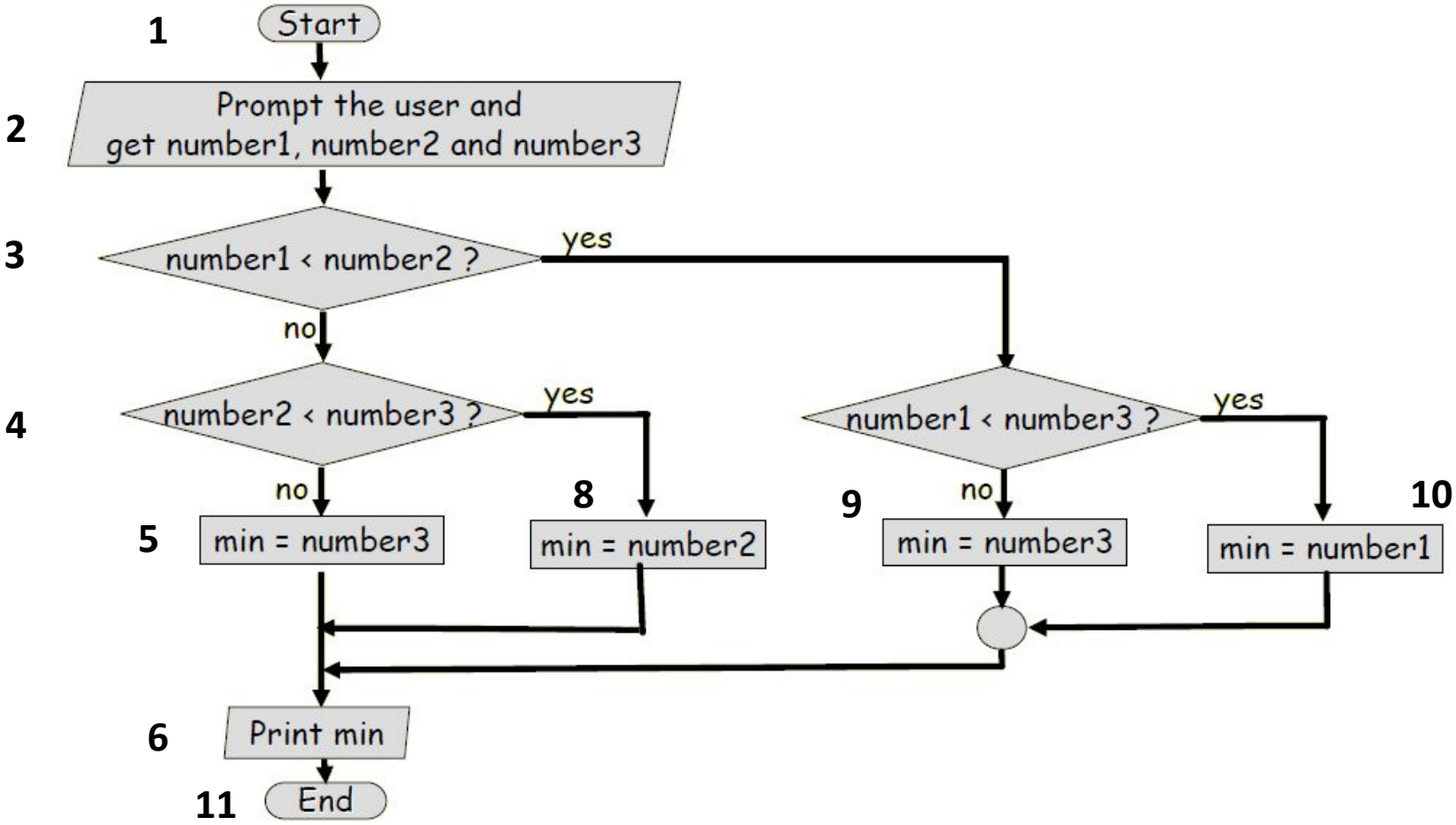
Ex6 - Compute the minimum of 3 numbers

- Problem: Design an algorithm to find the minimum of 3 numbers
- Input
  - 3 numbers
- Output
  - Minimum of the numbers

Ex6 - Compute the minimum of 3 numbers


```
1.    START
2.    Read n1, n2, n3       // consider n1, n2, n3 to be distinct
3.    if  n1 < n2
 3.1      if  n1 < n3
 3.1.1          min = n1
 3.2     else
 3.2.1    min = n3
 4.    else
 4.1       if n2 < n3
 4.1.1    min = n2
 4.2       else
 4.2.1    min = n3
 5.    Print min
 6.    STOP
```

Ex6- flowchart for compute the minimum of 3 numbers

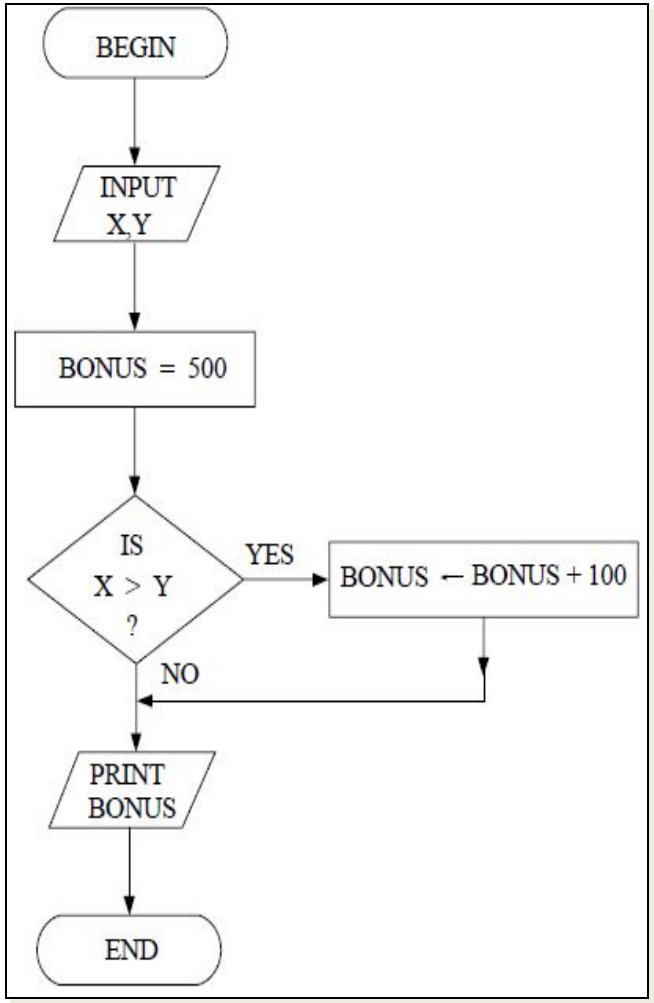


Another Approach

1. START
2. Read n1, n2, n3
3. min = n1 // (assume that n1 is min)
4. if n2 < min
   4.1 min = n2
5. if n3 < min
   5.1 min = n3
6. Print min
7. STOP

Can you tell the output?

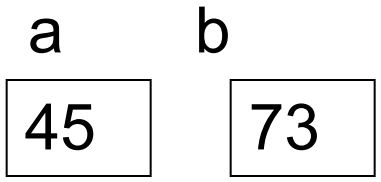
Consider the flowchart shown.
Give the value of BONUS under the following conditions:
(a) X = 20, Y = 10
(b) X = 10, Y = 20



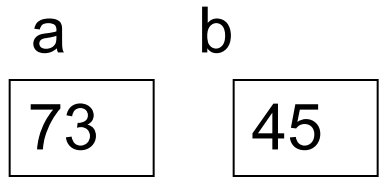
(a) BONUS = 600
(b) BONUS = 500

Try: Example Algorithm

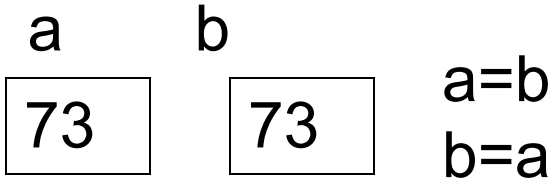
Exchanging/ Swapping the values of two variables

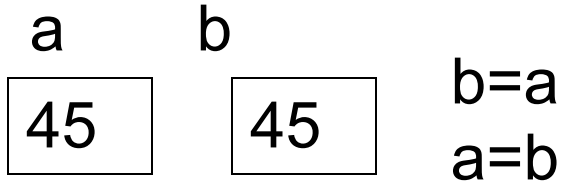


(Before Exchange)



(After Exchange)





✔ Let us try the solution using a third variable.

✔ Can you solve it without using any additional variable?

```c
#include<stdio.h>
 int main()
{
int a=10, b=20;
printf("Before swap a=%d b=%d",a,b);
a=a+b;//a=30 (10+20)
b=a-b;//b=10 (30-20)
a=a-b;//a=20 (30-10)
printf("\nAfter swap a=%d b=%d",a,b);
return 0;
}
```

```
Before swap a=10 b=20
After swap a=20 b=10

...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include<stdio.h>
 int main()
{
int a=10, b=20;
printf("Before swap a=%d b=%d",a,b);
a=a*b;//a=200 (10*20)
b=a/b;//b=10 (200/20)
a=a/b;//a=20 (200/10)
printf("\nAfter swap a=%d b=%d",a,b);
return 0;
}
```

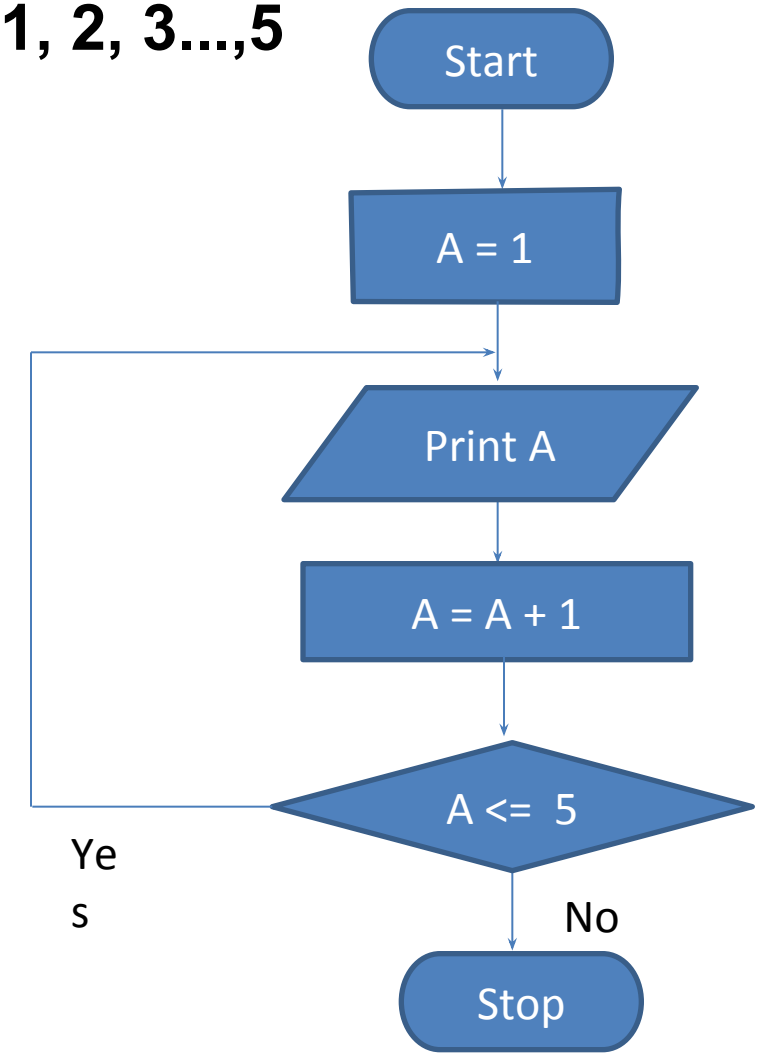
```
Before swap a=10 b=20
After swap a=20 b=10

...Program finished with exit code 0
Press ENTER to exit console.
```

Loops: Repetitive operations

Draw a flowchart for printing values 1, 2, 3...,5

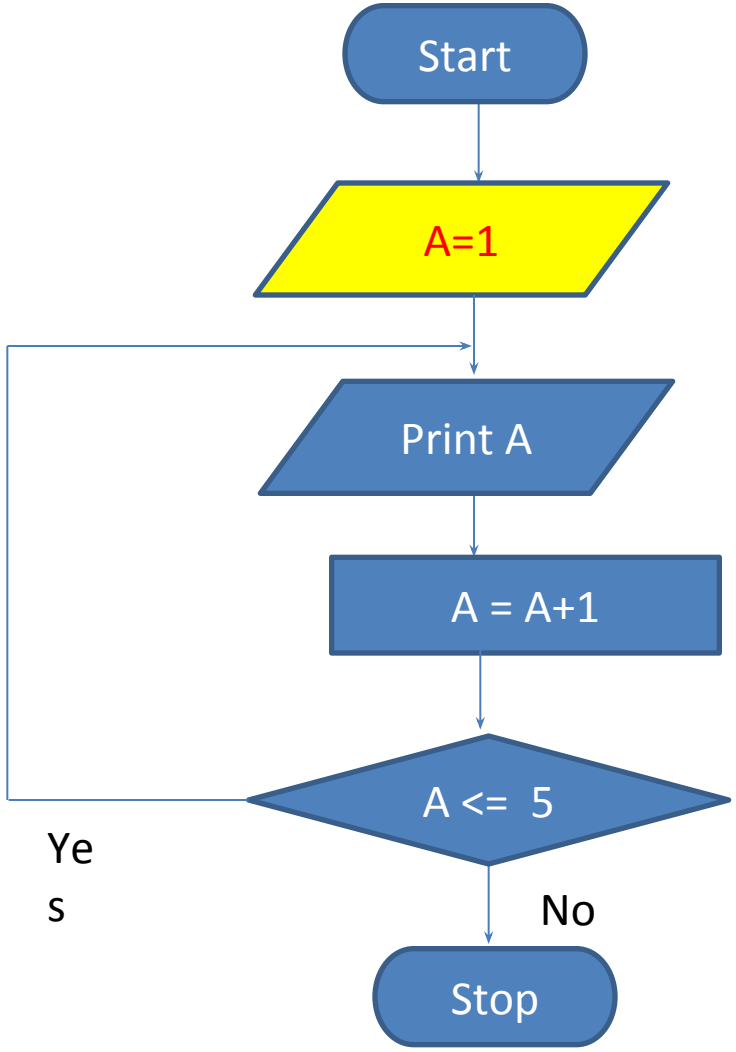
- **Initialization**
- **The Loop Variable**
- **Loop Exit Test**



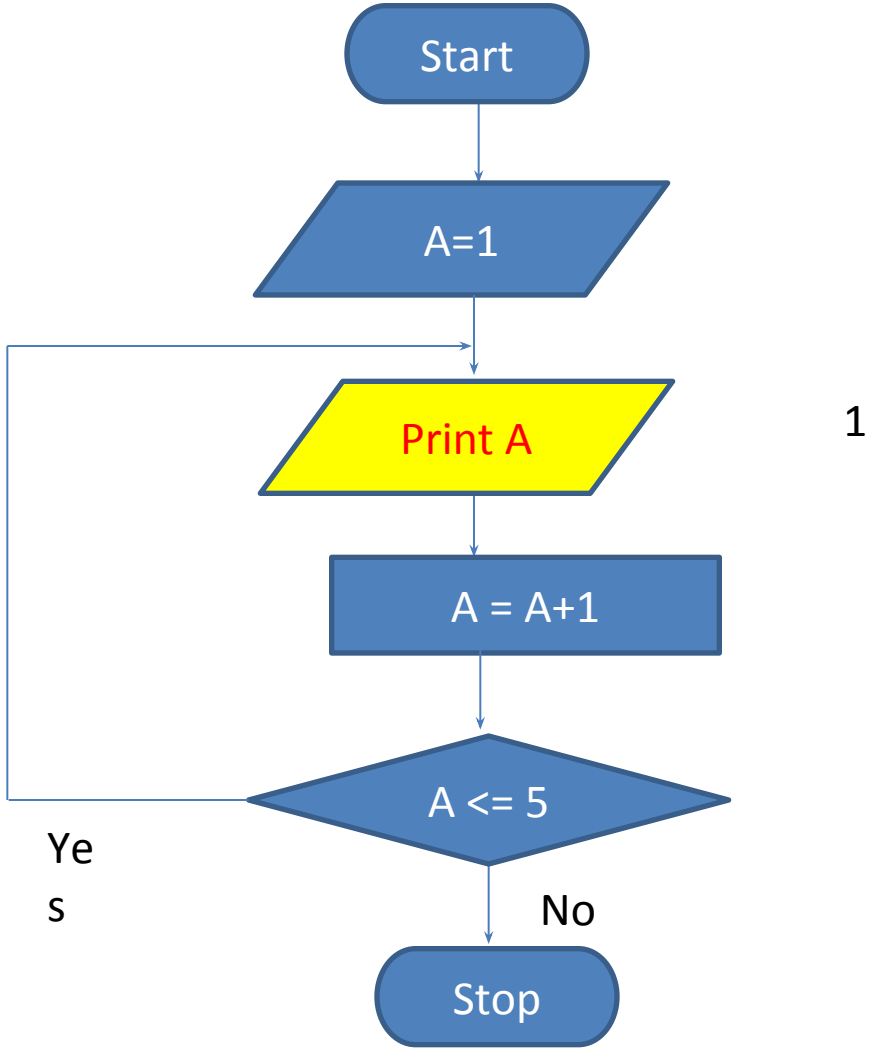
Algorithm

1. START
2. for A = 1 to 5
   2.1 Print A
3. STOP

Loops: Repetitive operations

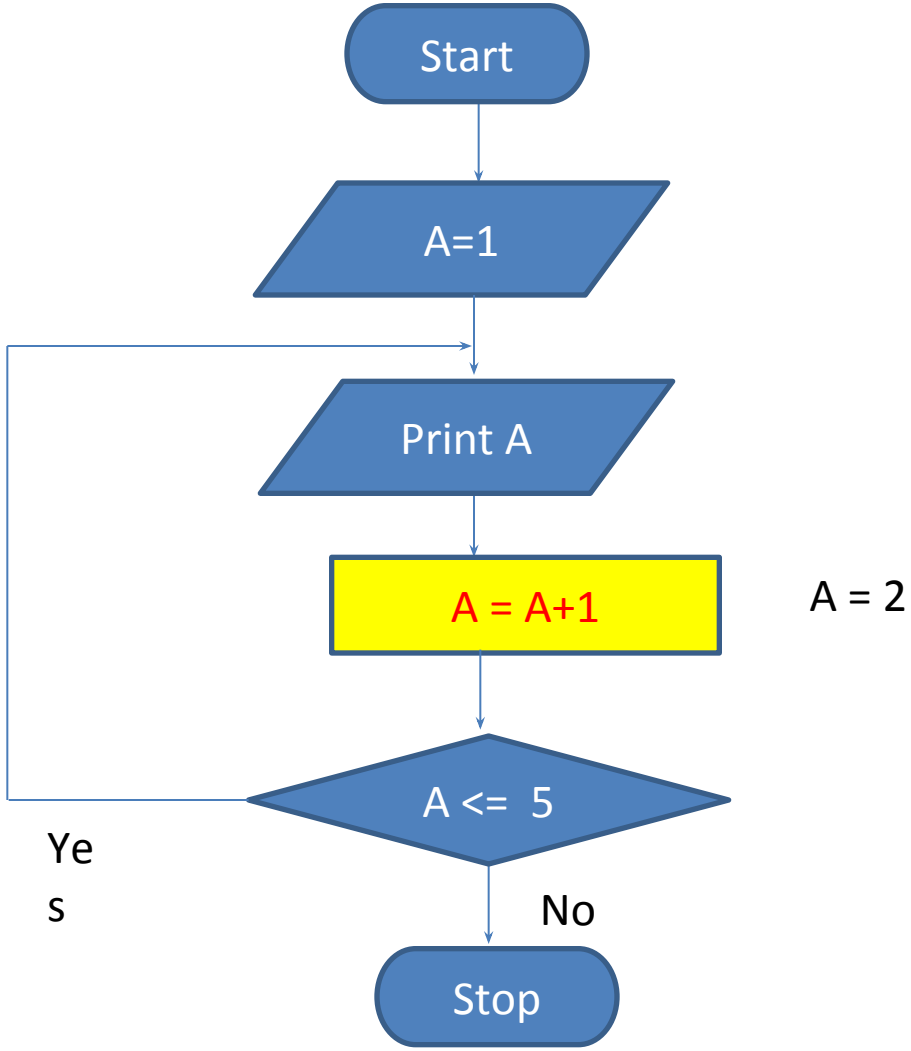


Loops: Repetitive operations

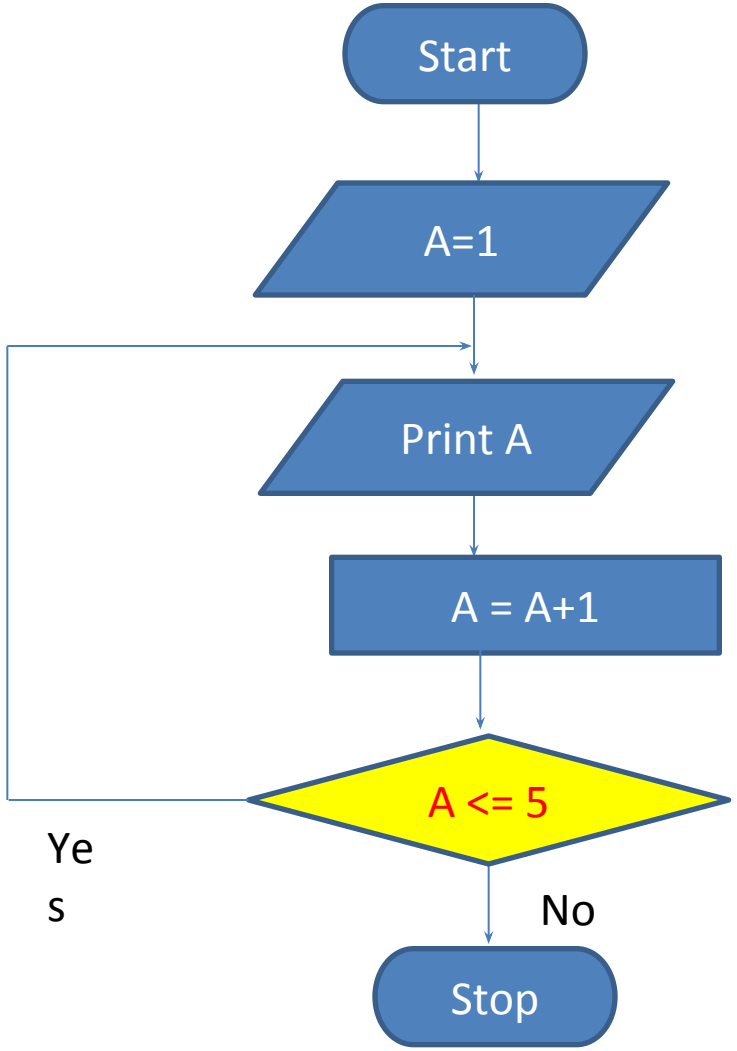


1

Loops: Repetitive operations

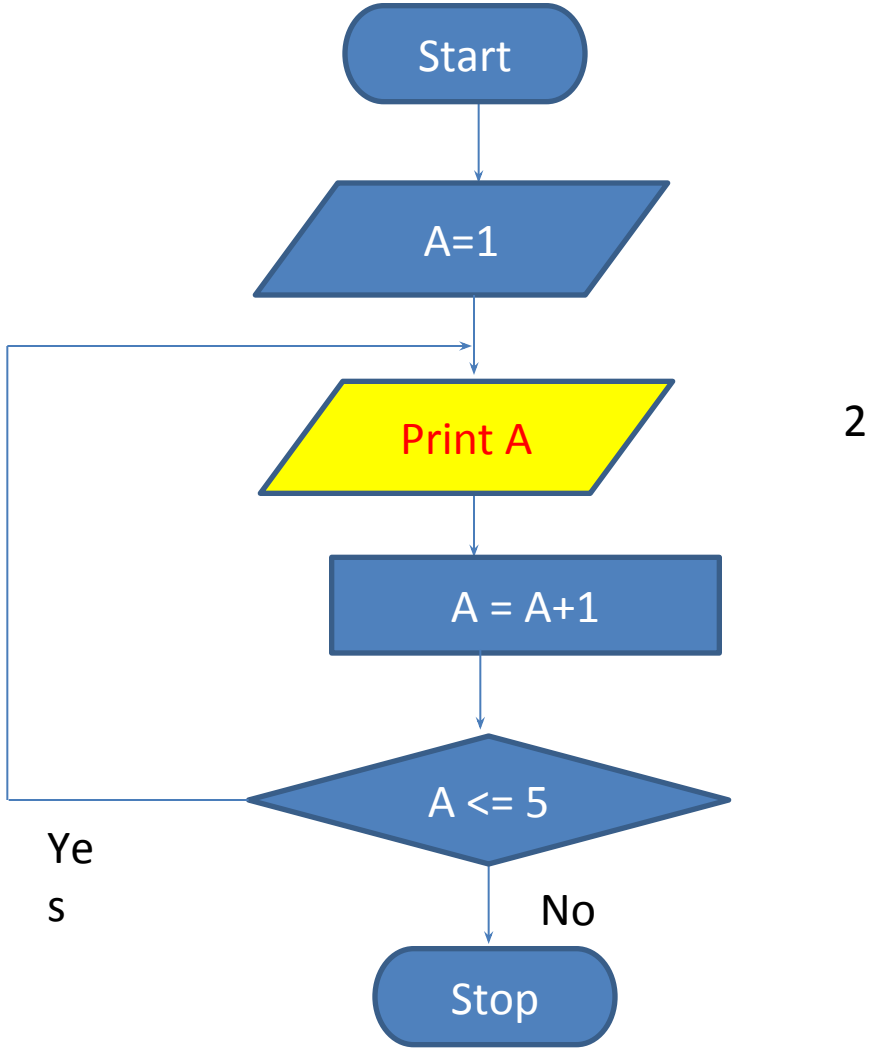


Loops: Repetitive operations



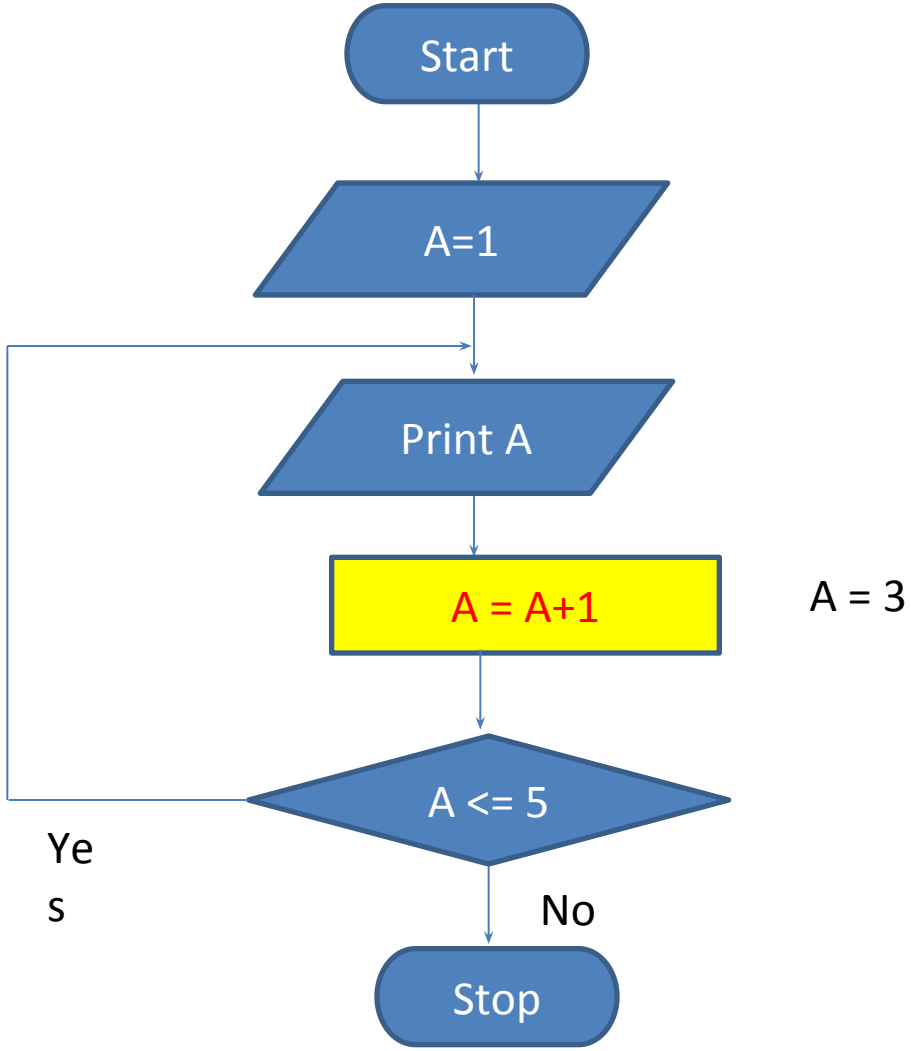
2 < 5 (Yes path is followed)

Loops: Repetitive operations



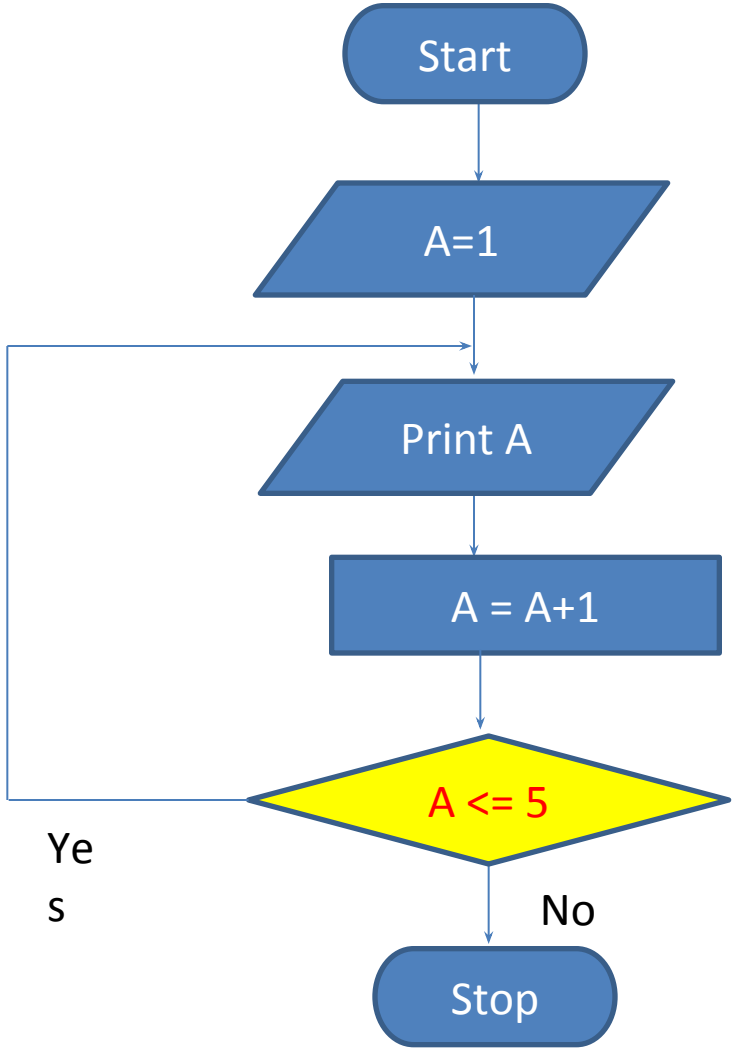
2

Loops: Repetitive operations



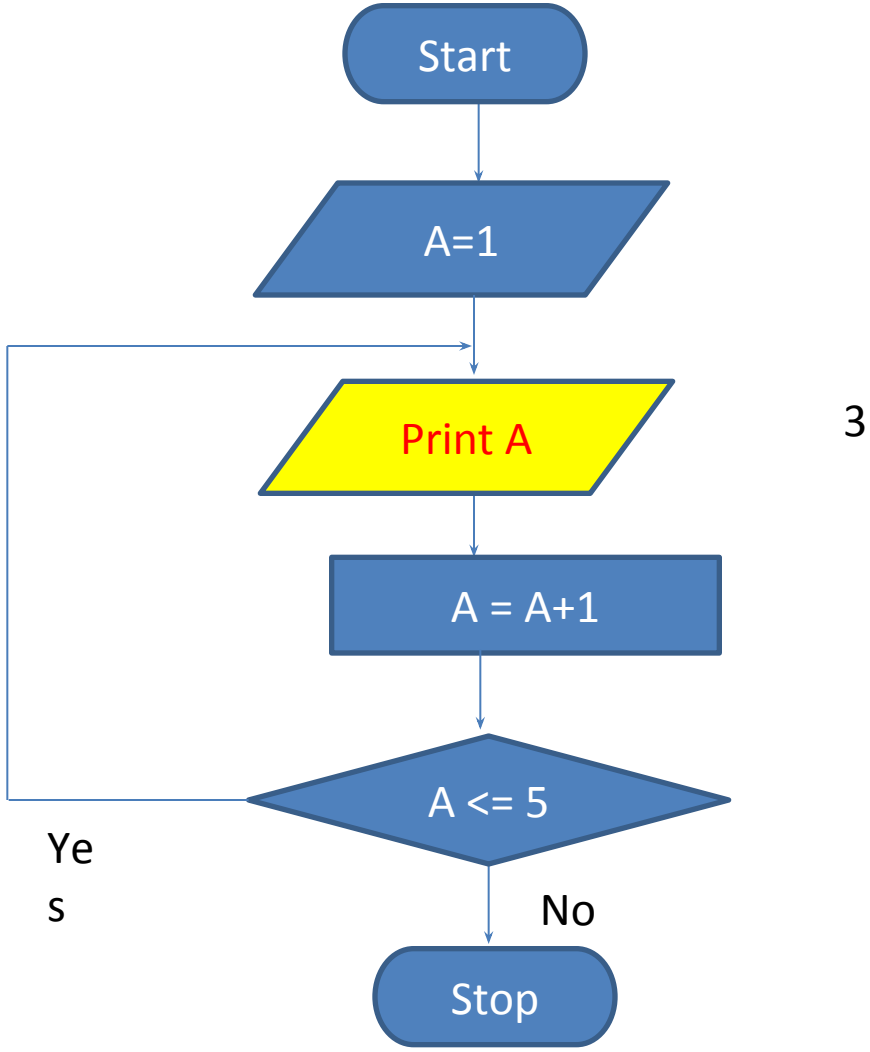
A = 3

Loops: Repetitive operations



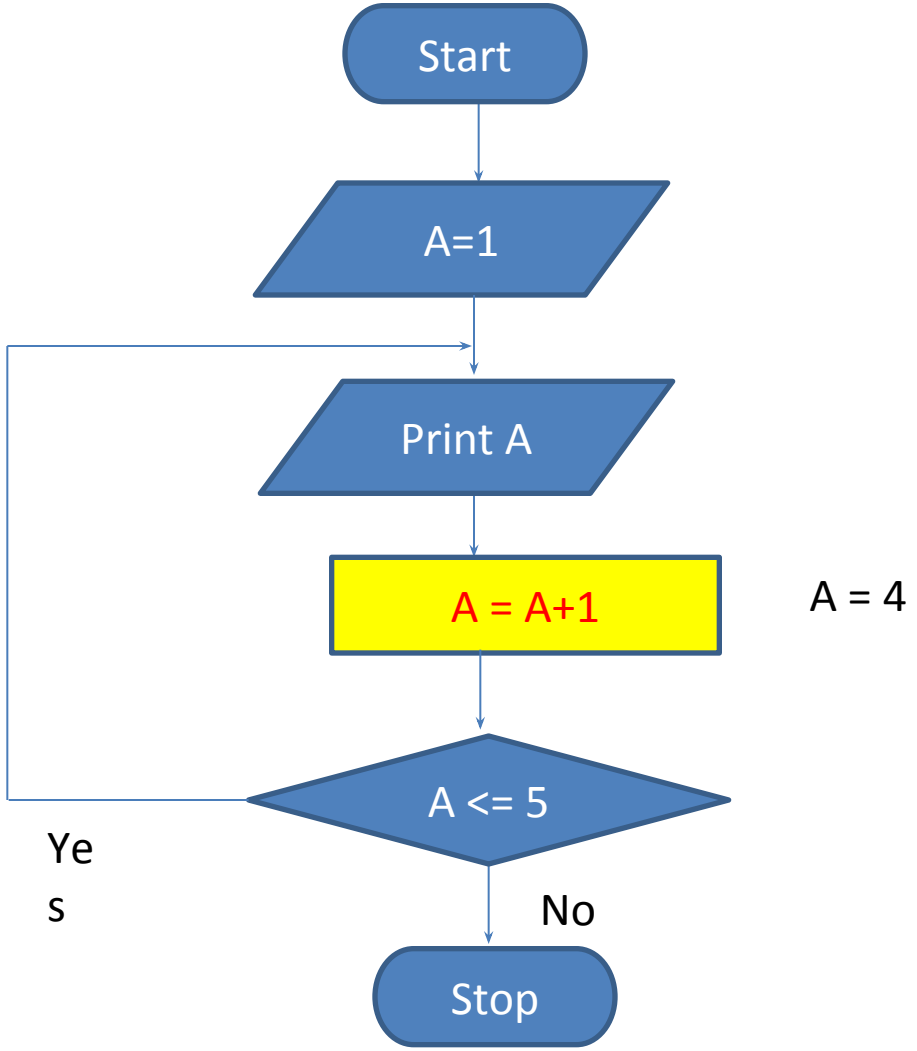
3 < 5 (Yes path is followed)

Loops: Repetitive operations

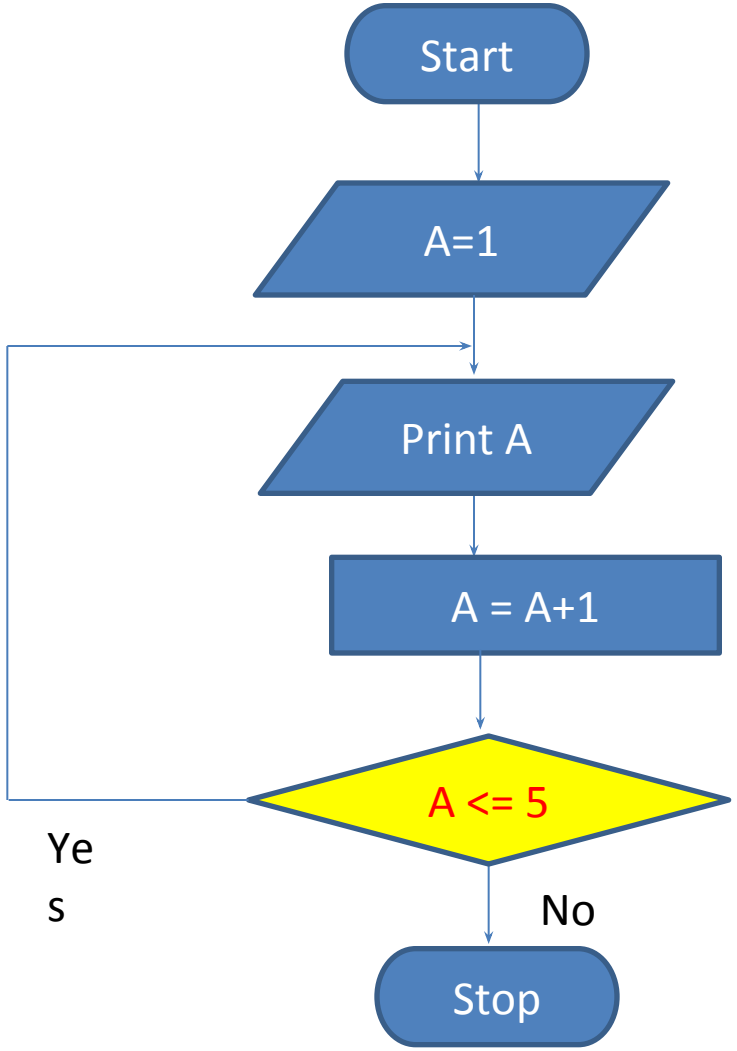


3

Loops: Repetitive operations

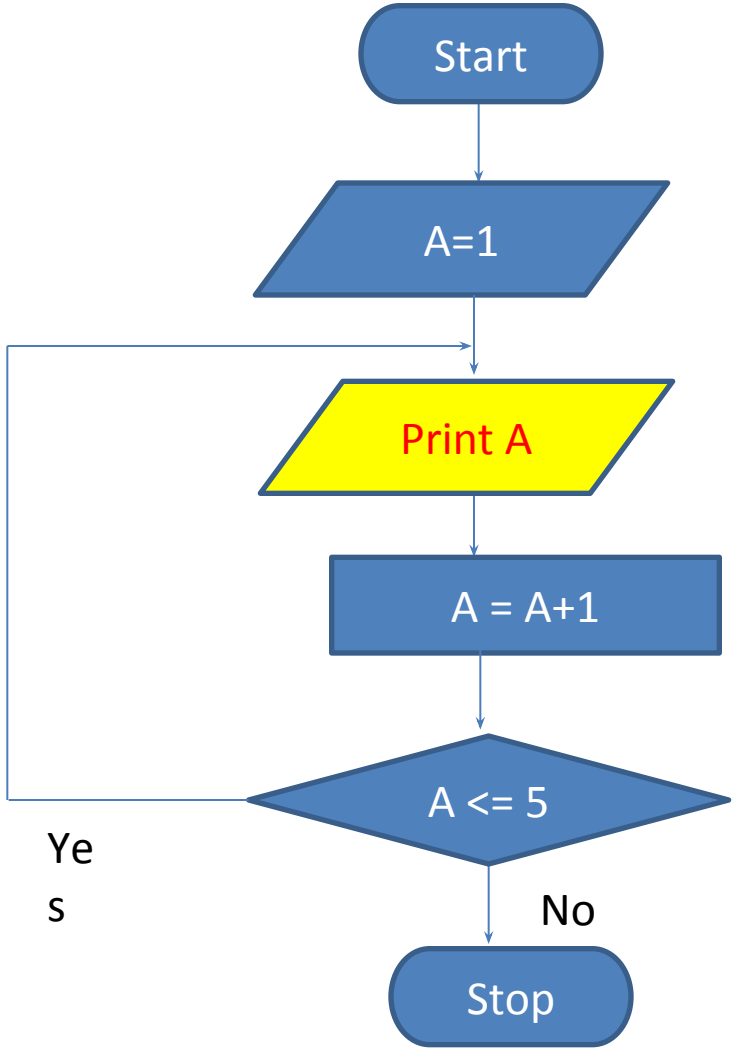


Loops: Repetitive operations



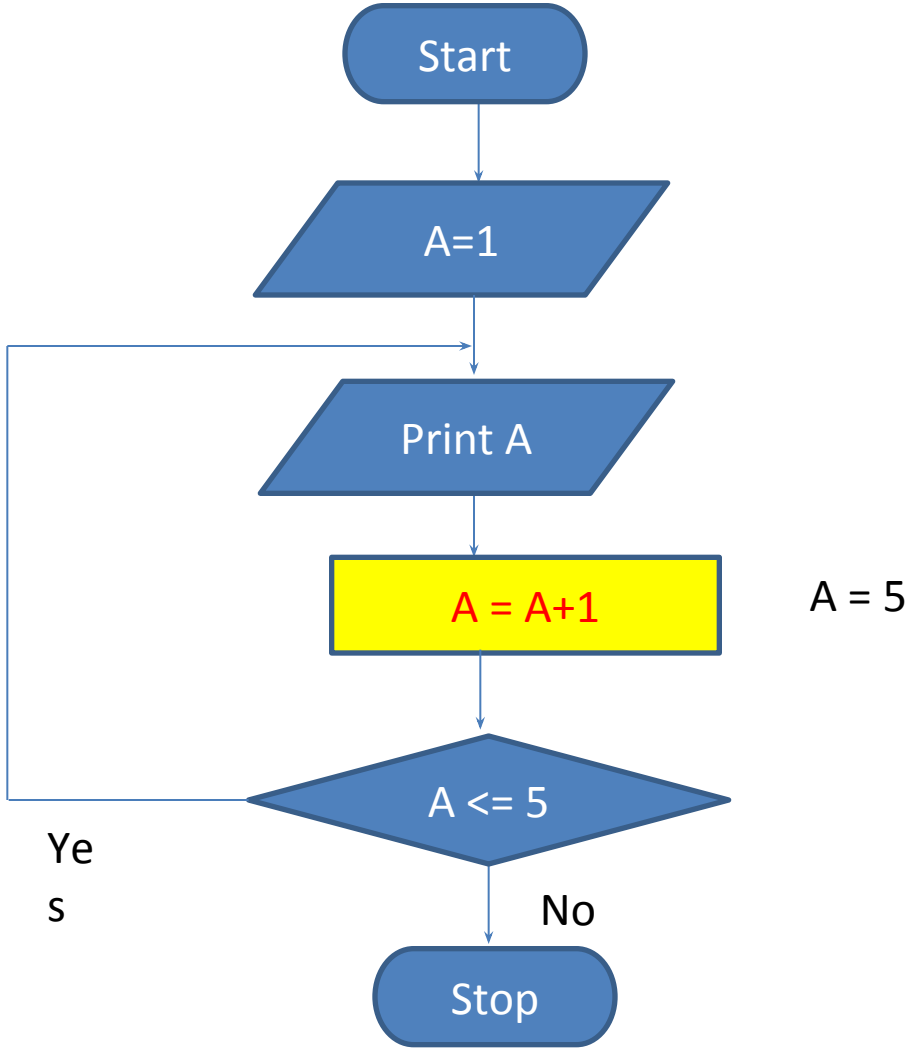
4 < 5 (Yes path is followed)

Loops: Repetitive operations

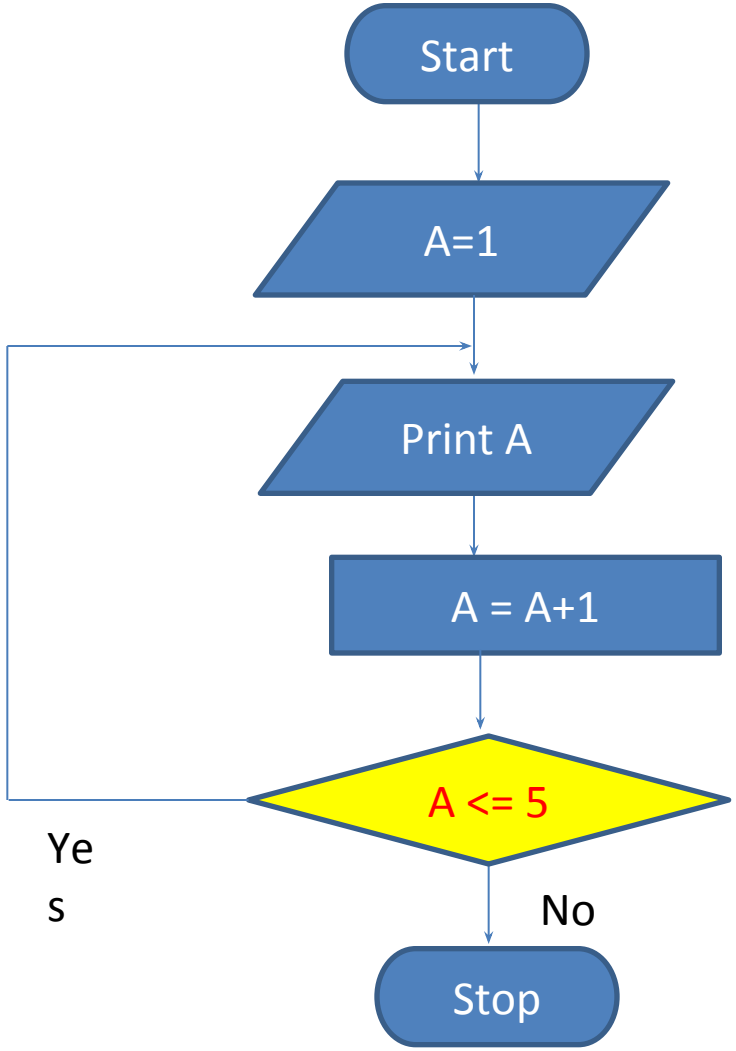


4

Loops: Repetitive operations

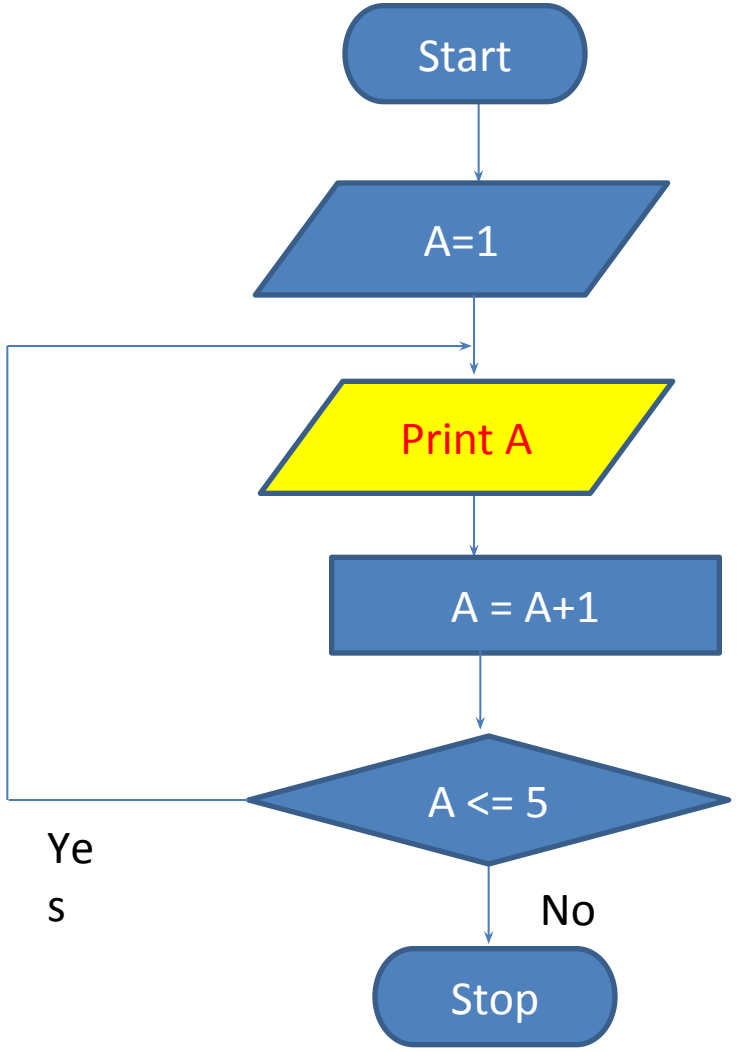


Loops: Repetitive operations



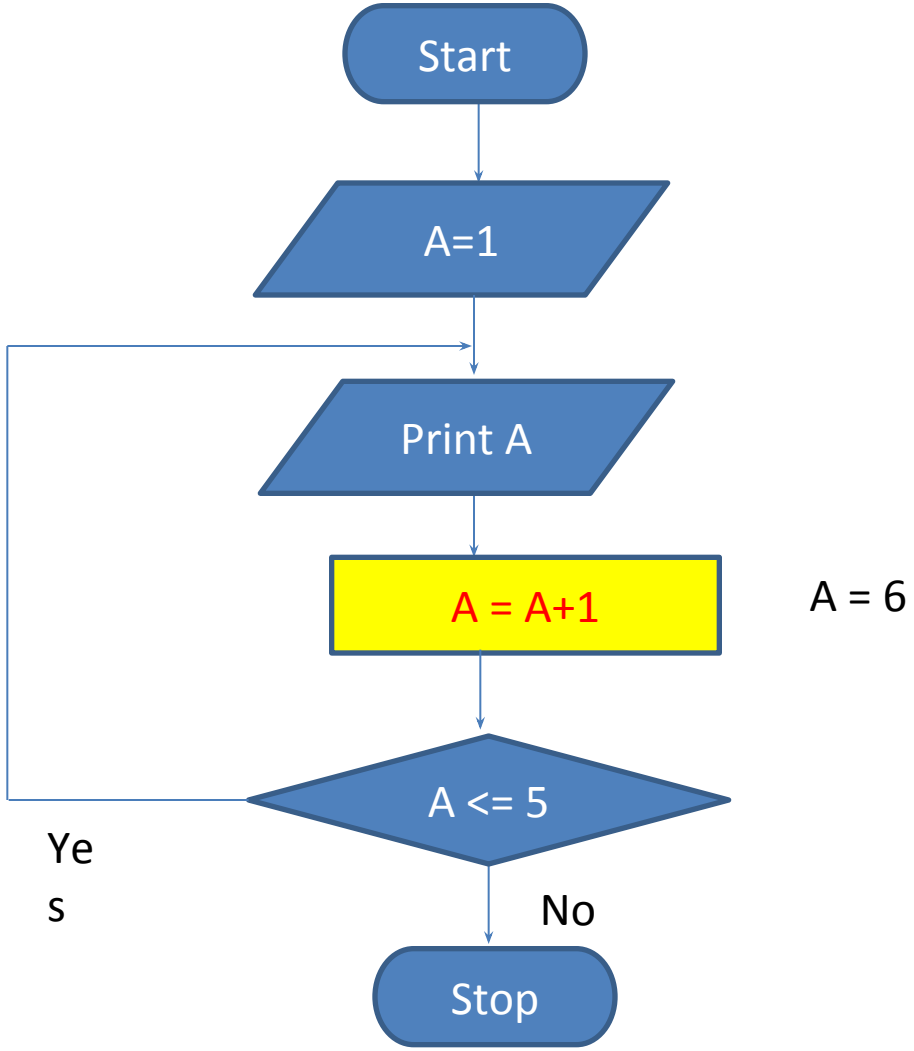
5 = 5 (Yes path is followed)

Loops: Repetitive operations

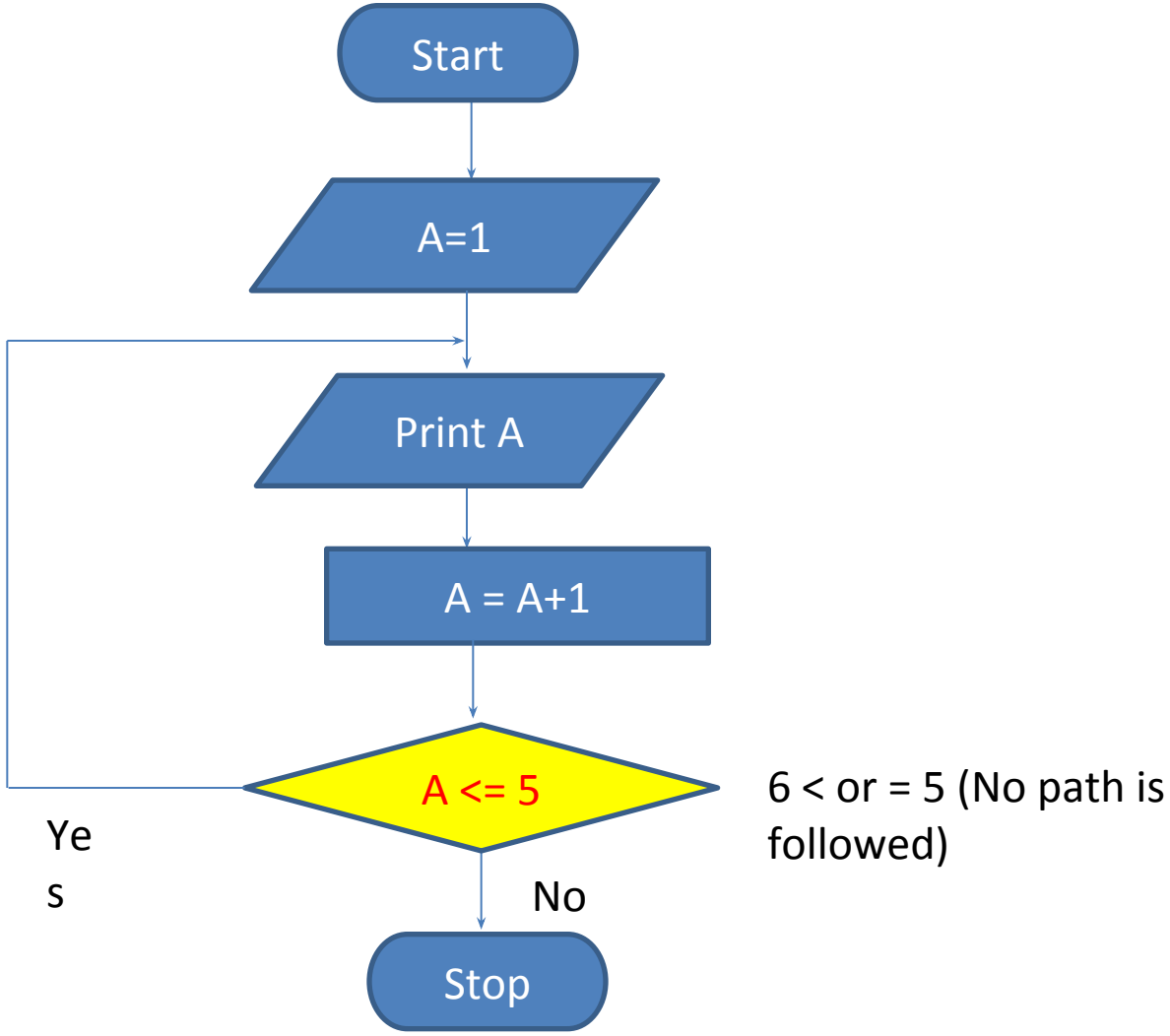


5

Loops: Repetitive operations



Loops: Repetitive operations



6 < or = 5 (No path is followed)

What is printed by the flowcharts, shown in Figures 1 and 2.

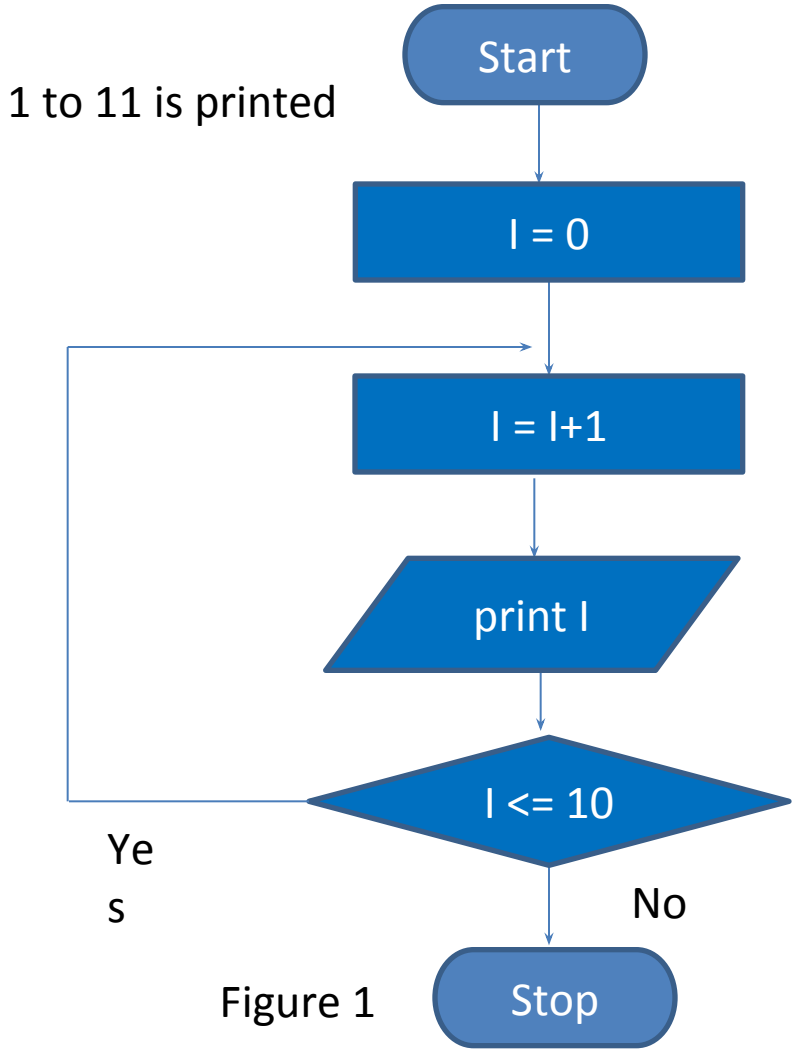


Figure 1

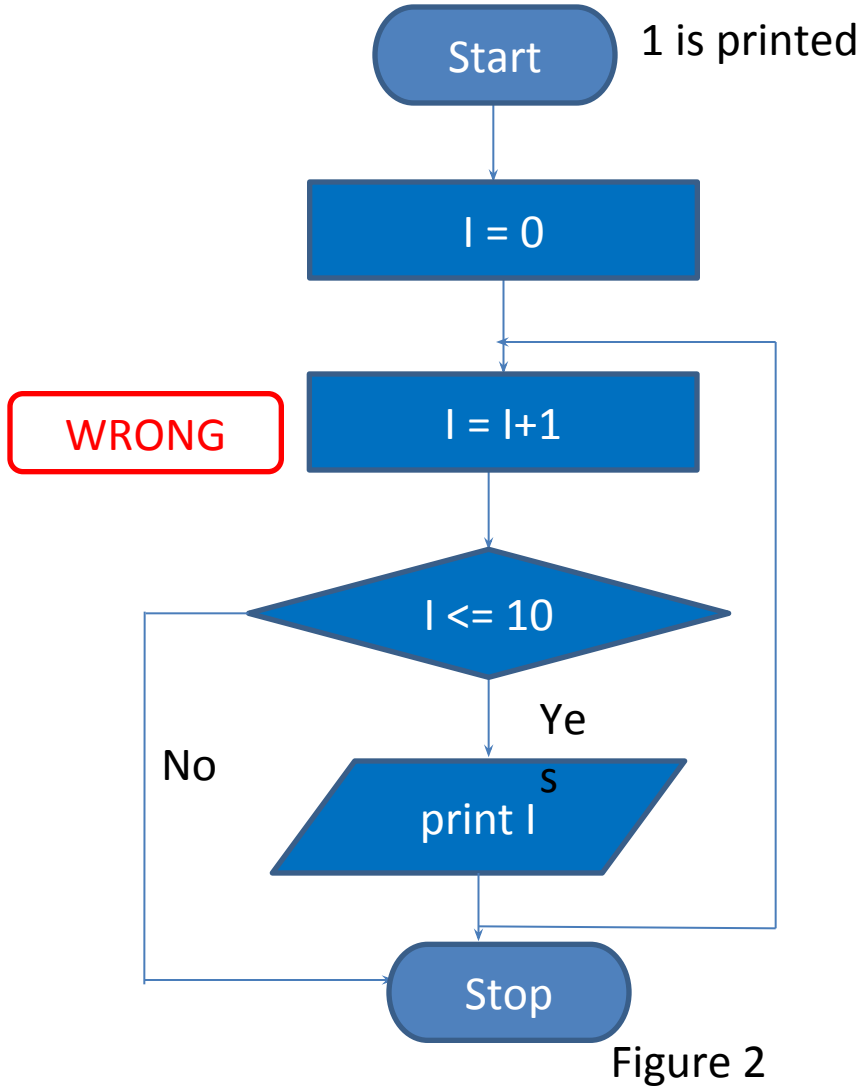


Figure 2

Figure-1

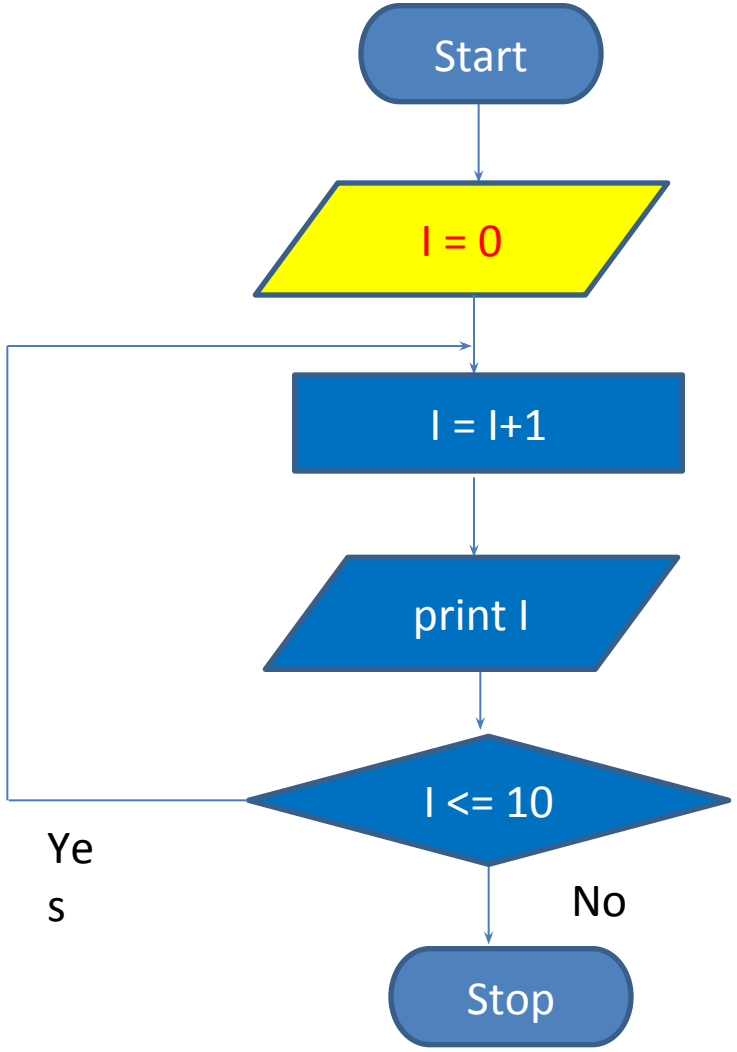


Figure-1

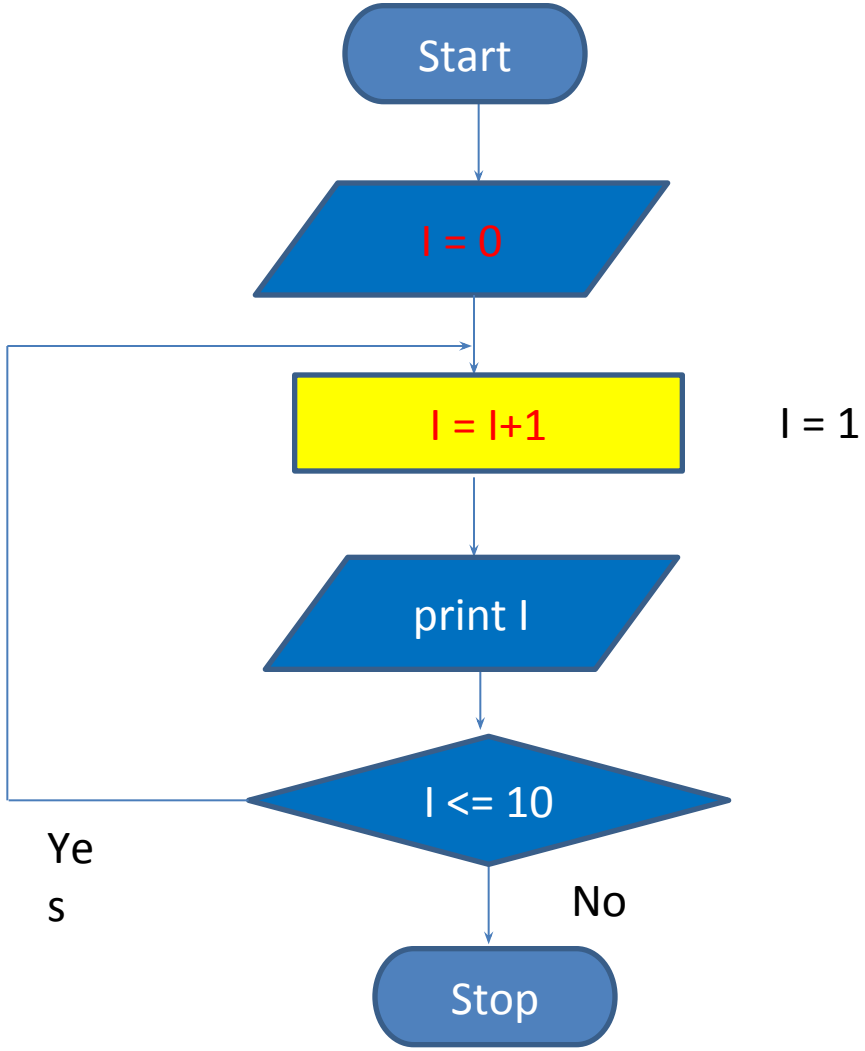


Figure-1

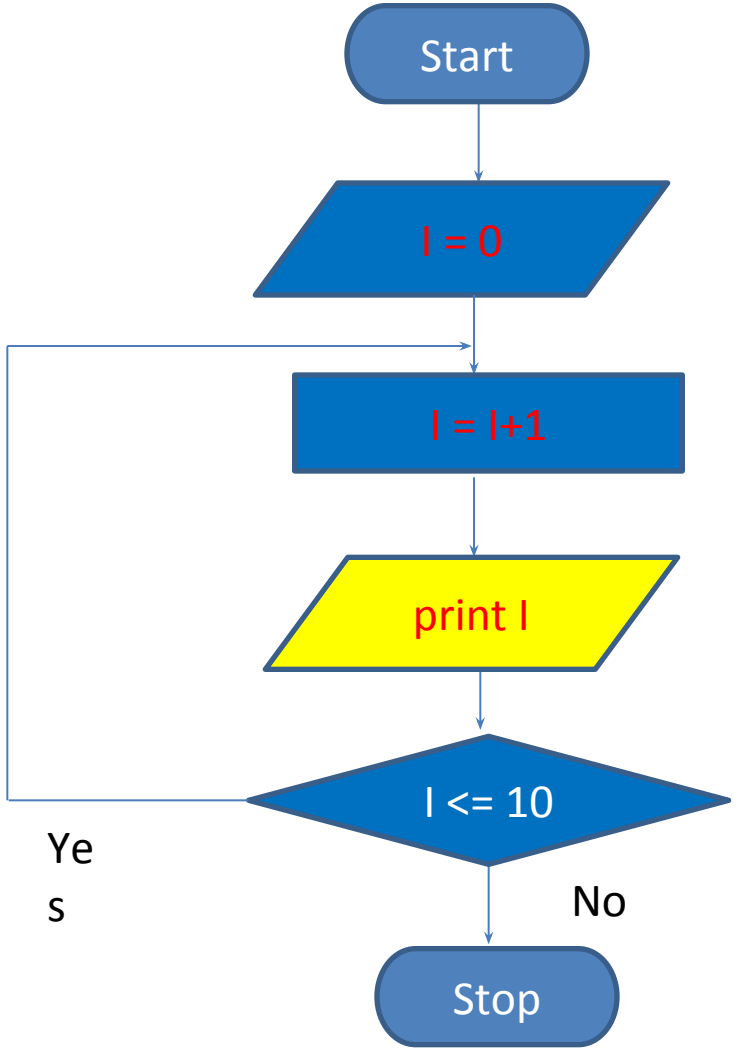


Figure-1

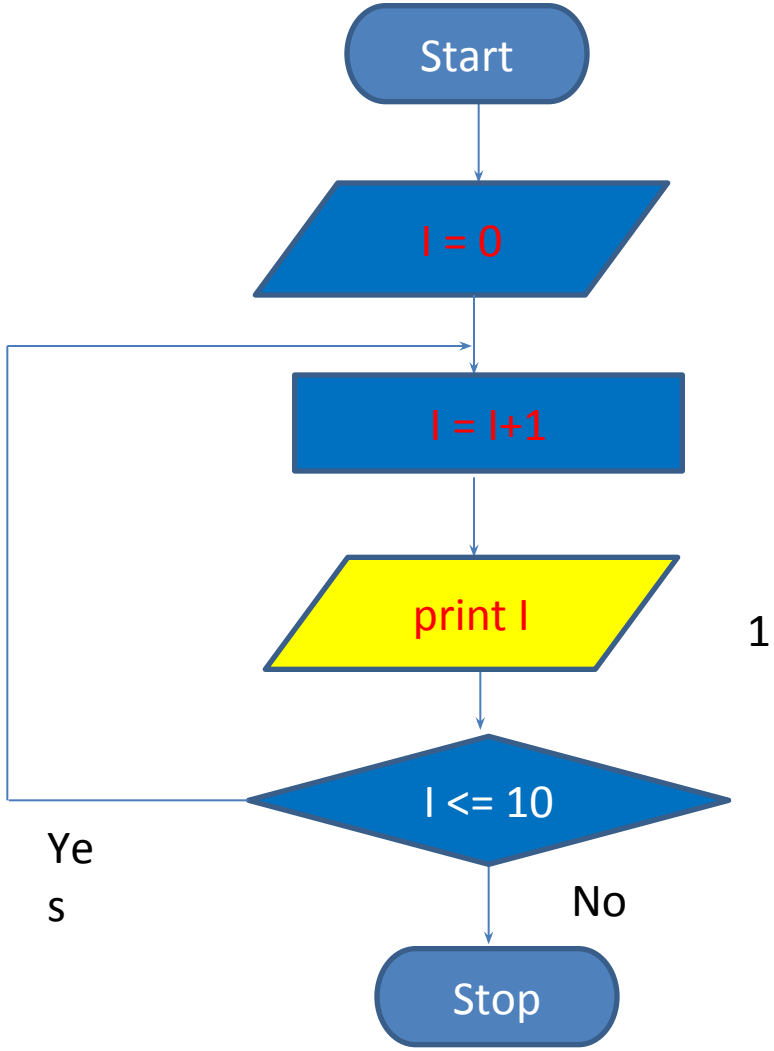
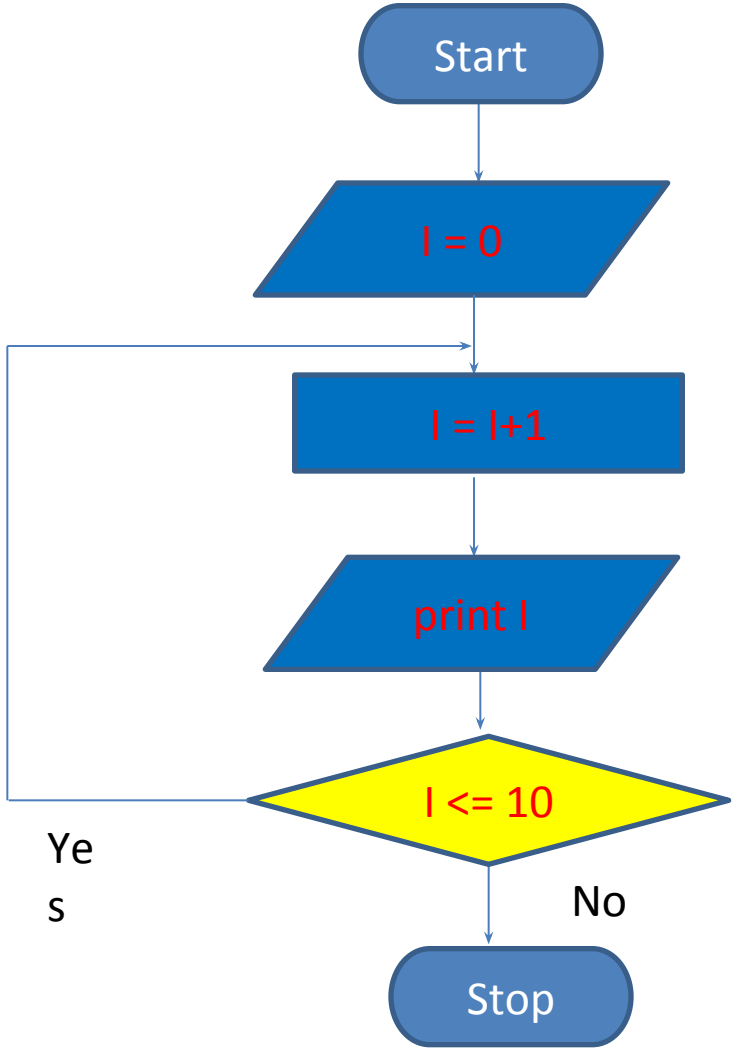


Figure-1



1 < 10 Yes Path followed

Figure-1

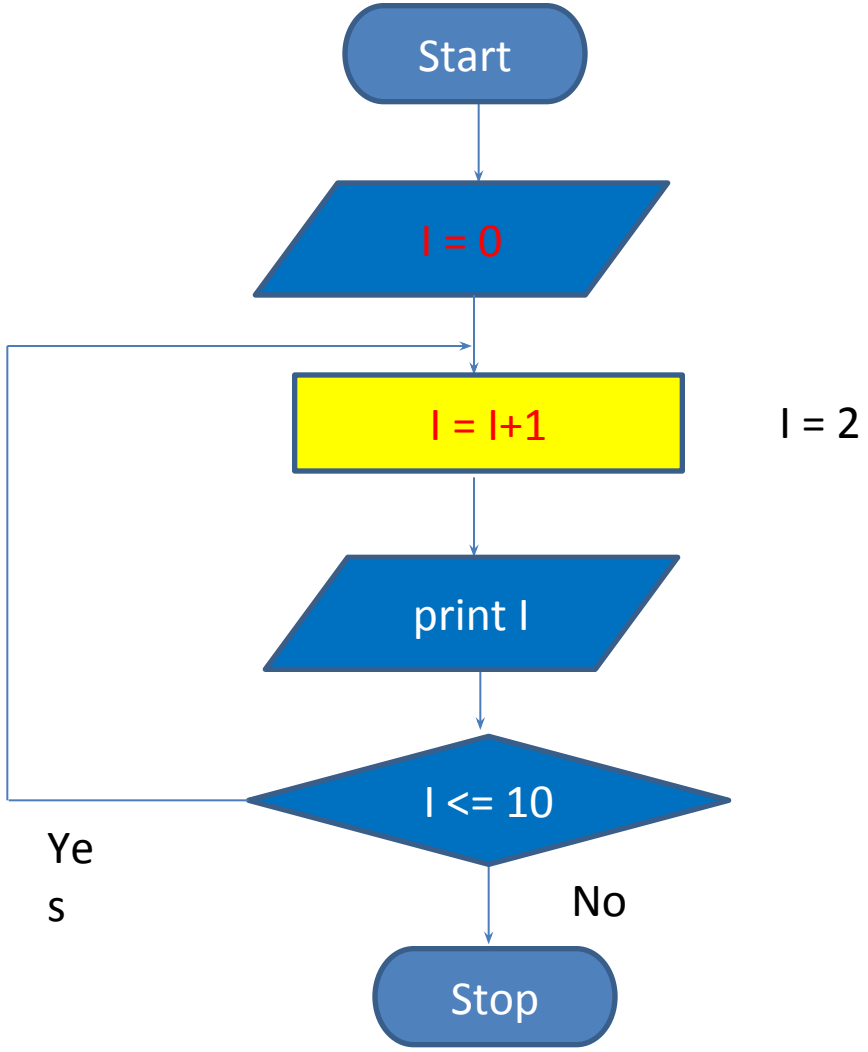


Figure-1

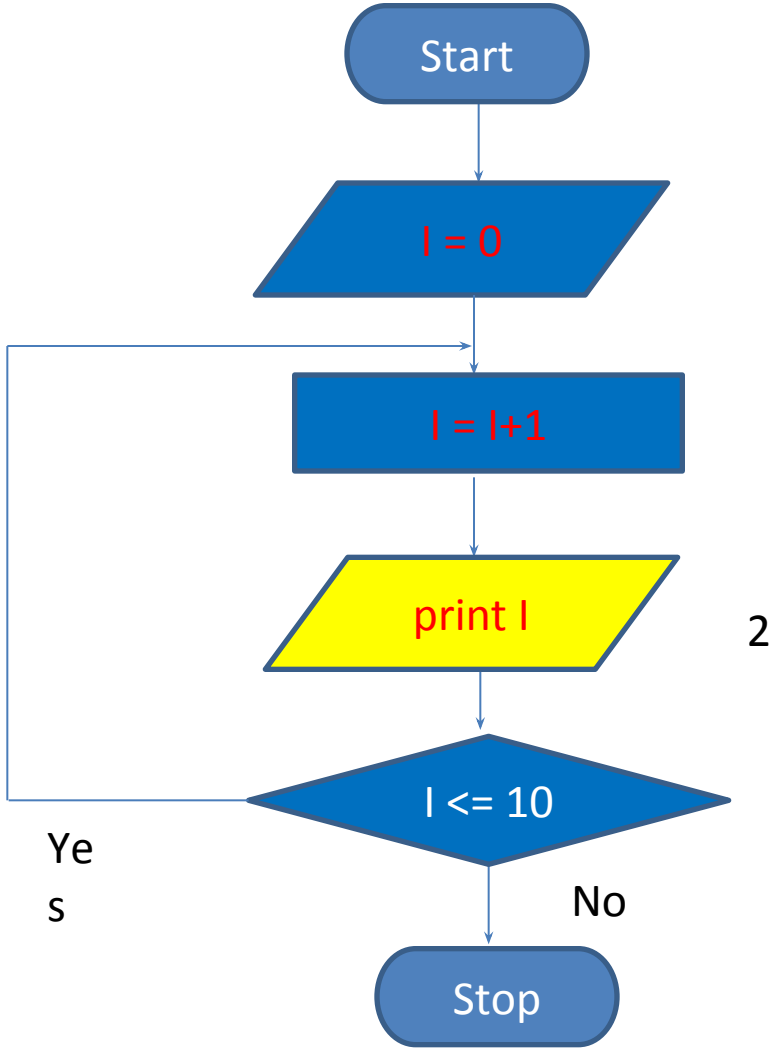
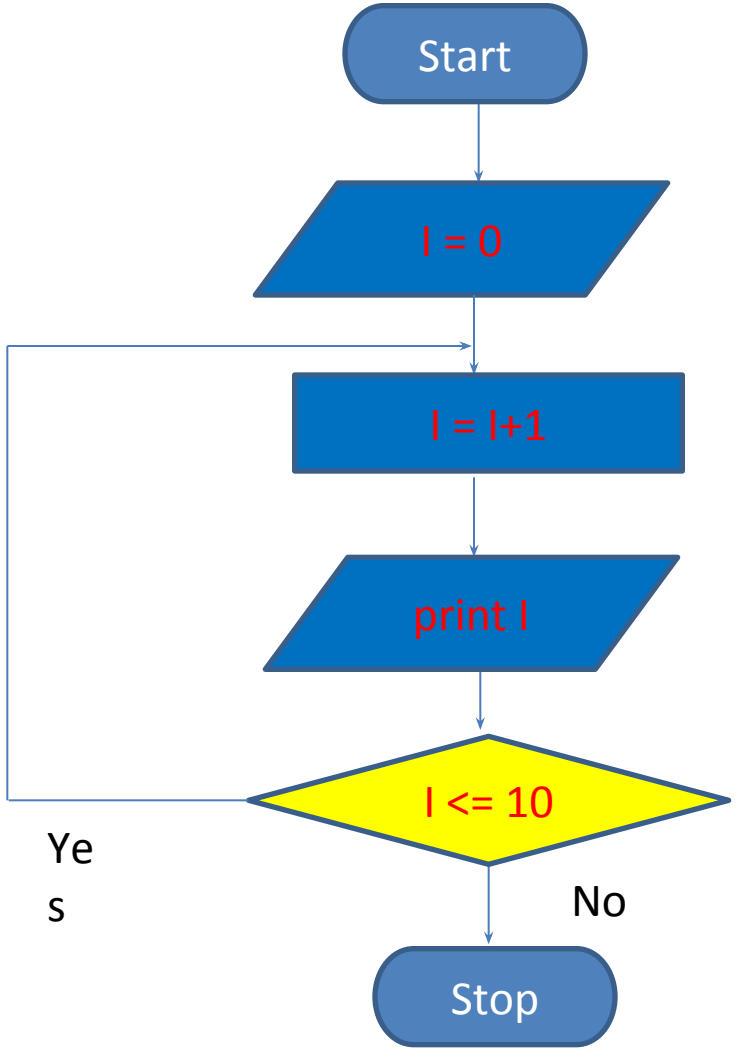
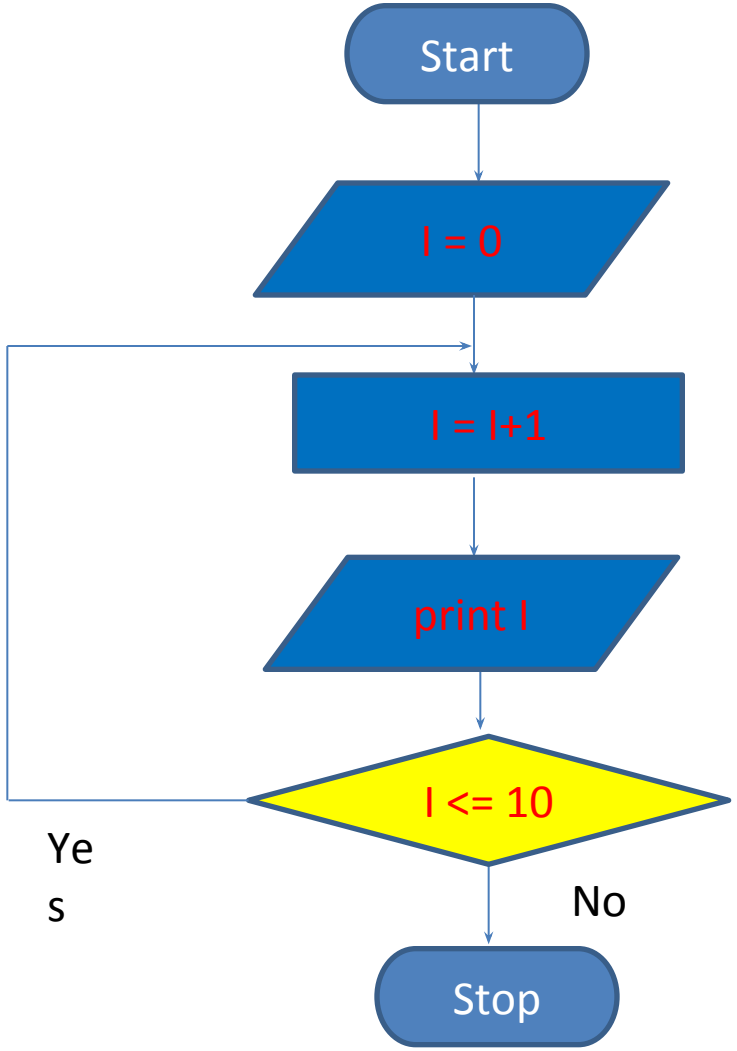


Figure-1



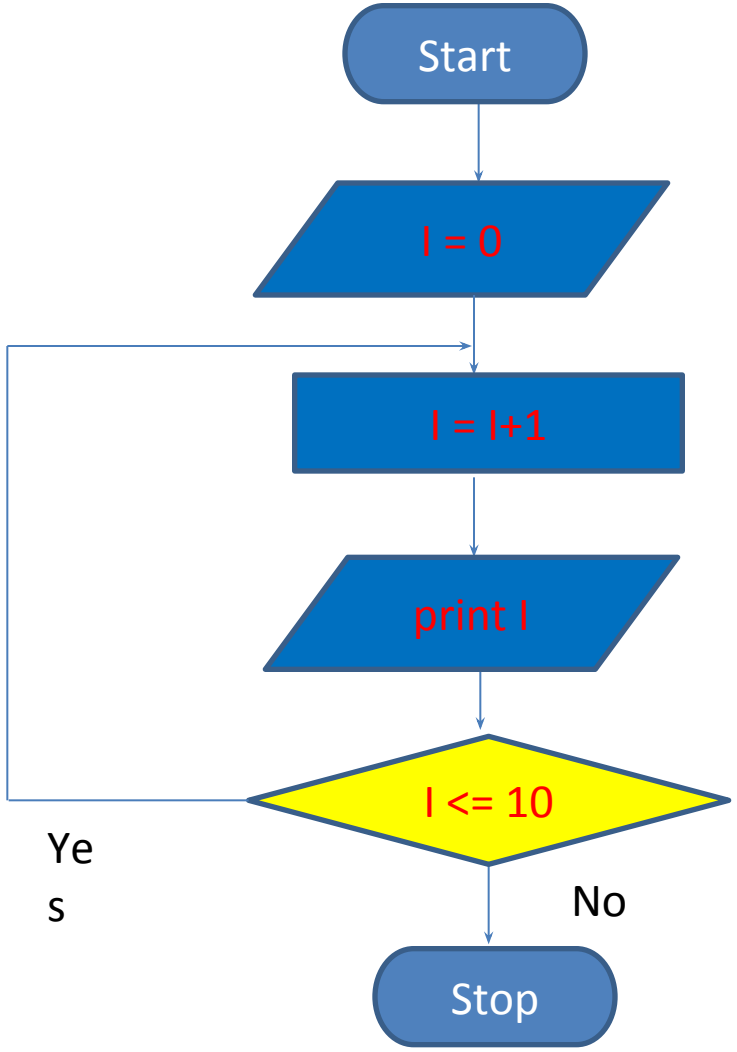
2 < 10 Yes Path followed

Figure-1



Continuing the increment
In this way when I reaches 10

Figure-1



10 <= 10 Yes Path followed

Figure-1

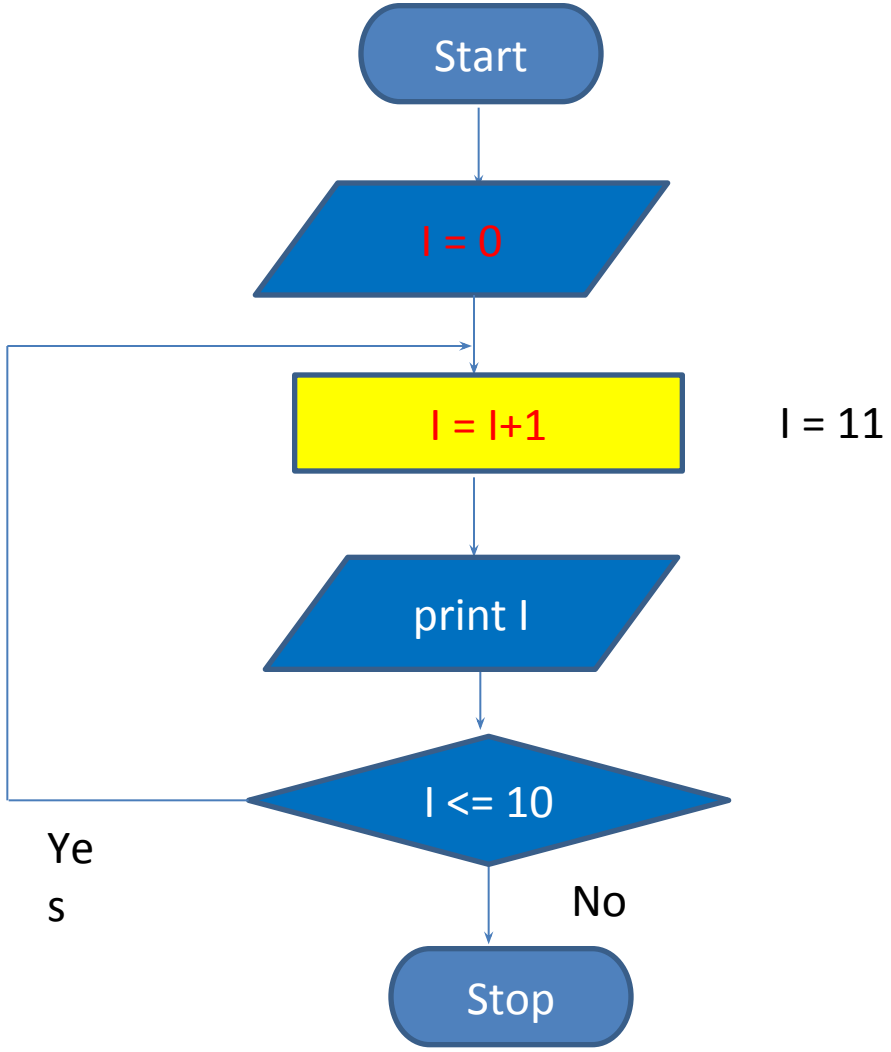


Figure-1

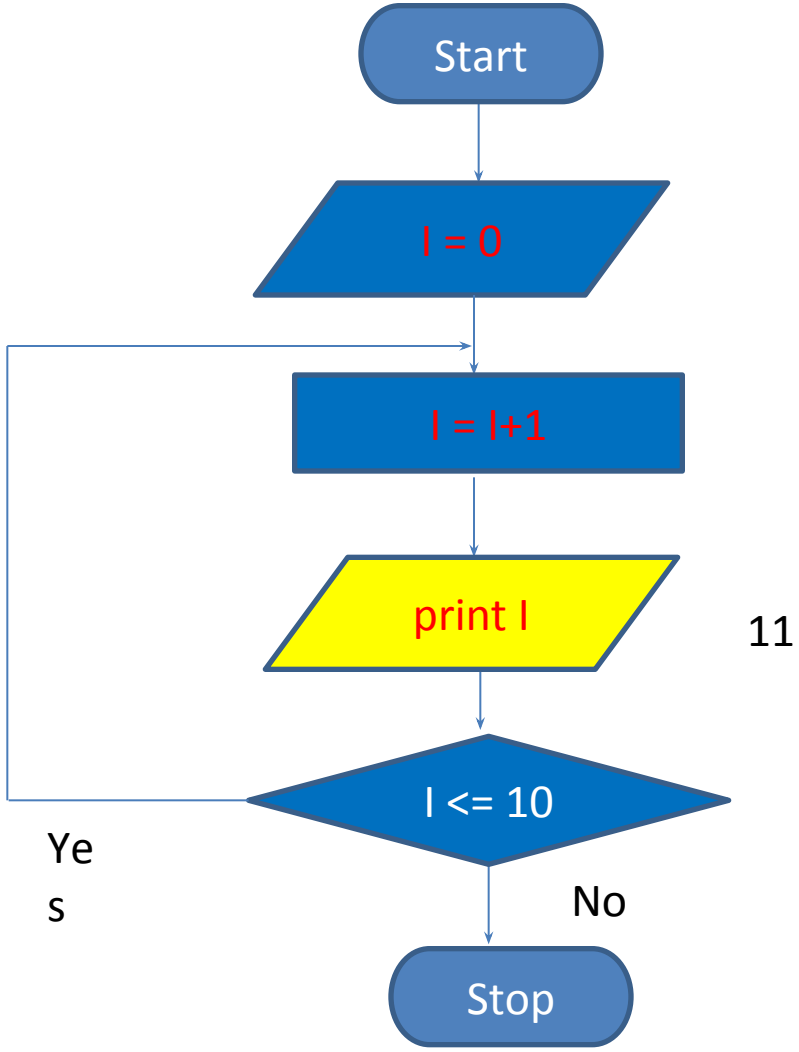
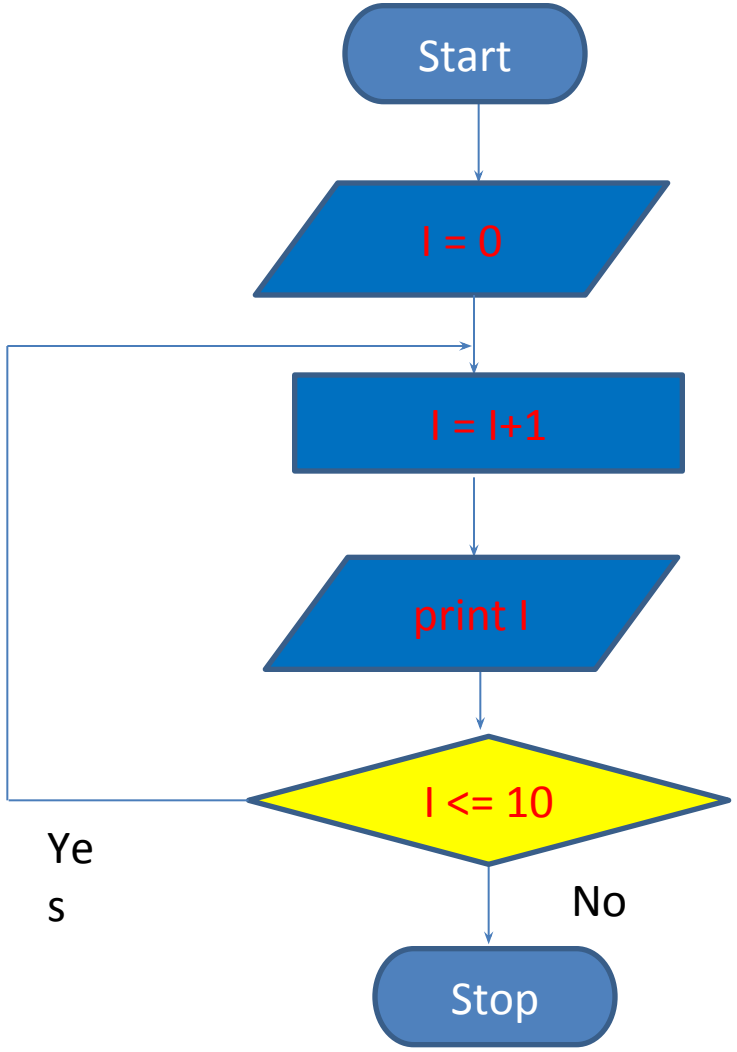
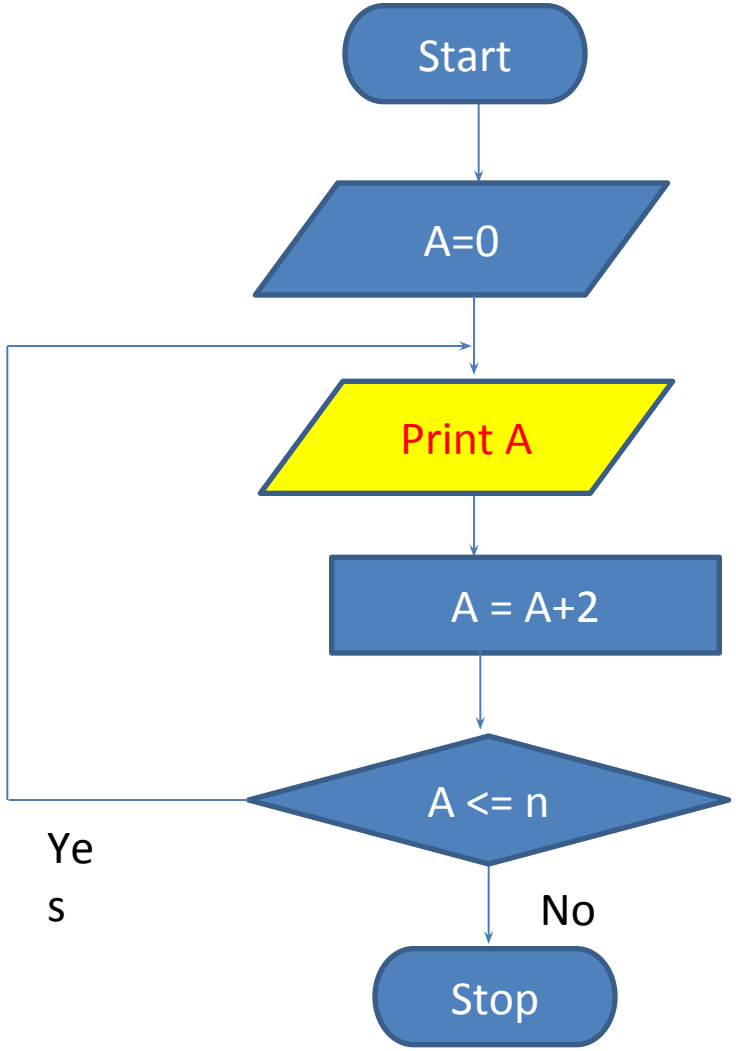


Figure-1

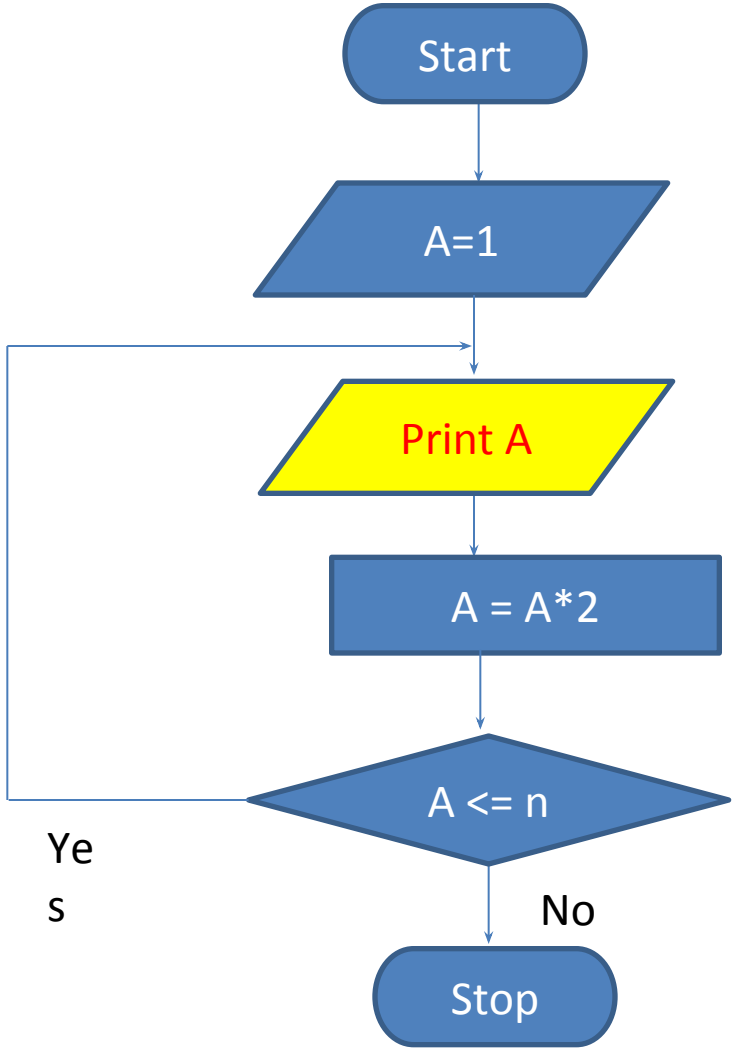


11 < 10 No Path followed

Loops: Repetitive operations



Loops: Repetitive operations



Sum 1 + 2 + 3 … + N

1. START
2. Read N
3. sum = 0
4. for i = 1 to N

4.1 sum = sum + i

5. print sum
6. STOP

Sum of Odd numbers till N

1. START
2. Read N
3. sum = 0
4. for i = 1 to N step +2

4.1 sum = sum + i

5. print sum
6. STOP

Even numbers 20, 18, 16 … 0

1. START
2. for i = 20 to 0 step -2

2.1 print i

3. STOP