ALGORITHMS & FLOWCHARTING II

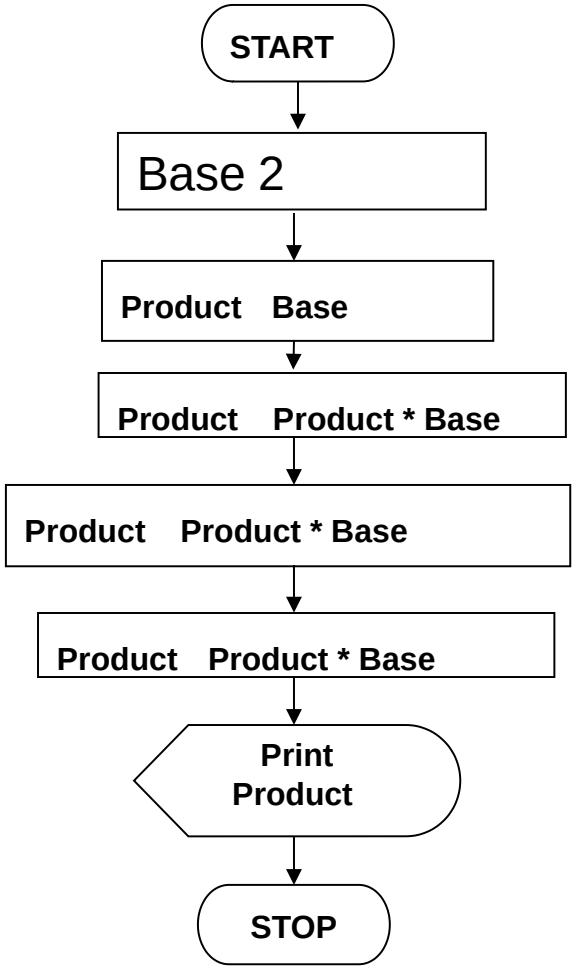
LOOPS

- Computers are particularly well suited to applications in which operations are repeated many times.
- If the same task is repeated over and over again a loop can be used to reduce program size and complexity

Example 8: Write an algorithm and draw a flowchart to calculate 2^4 .

- **Algorithm**:
- Step 1: Base 2
- Step 2: Product Base
- Step 3: Product Product * Base
- Step 4: Product Product * Base
- Step 5: Product Product * Base
- Step 6: *Print* Product

Flowchart

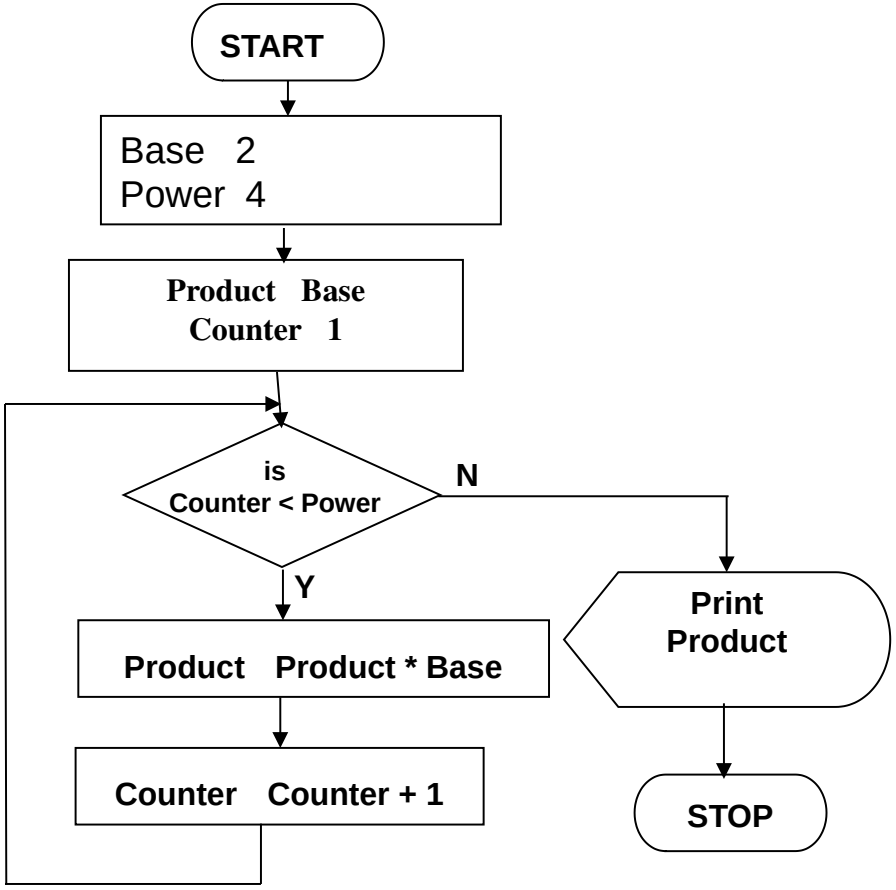

START
Base 2
Product Base
Product Product * Base
Product Product * Base
Product Product * Base
Print Product
STOP

- Question: What happens if you want to calculate 2 to the power of 1000?
- **Answer**: Use a LOOP (repeated execution of the same set of instructions)

- Example 9: Write an algorithm and draw a flowchart to calculate 2^4 using a loop approach? Verify your result by a *trace table*.

Algorithm:

Step 1: Base 2
Step 2: Power 4
Step 3: Product Base
Step 4: Counter 1
Step 5: While Counter < Power
Repeat Step 5 through step 7
Step 6: Product Product * Base
Step 7: Counter Counter +1
Step 8: *Print* Product


START
Base 2
Power 4
Product Base
Counter 1
is
Counter < Power
N
Y
Print
Product
Product Product * Base
Counter Counter + 1
STOP

TRACING

	BASE	POWER	PRODUCT	COUNTER	COUNTER < POWER
STEP 1:	2	?	?	?	?
STEP 2:	2	4	?	?	?
STEP 3:	2	4	2	?	?
STEP 4:	2	4	2	1	T
STEP 5:	**2**	**4**	**2**	**1**	**T**
STEP 6:	2	4	2x2=4	1	T
STEP 7:	2	4	4	1+1=2	T
STEP 5:	**2**	**4**	**4**	**2**	**T**
STEP 6:	2	4	4x2=8	2	T
STEP 7:	2	4	8	2+1=3	T
STEP 5:	**2**	**4**	**8**	**3**	**T**
STEP 6:	2	4	8x2=16	3	T
STEP 7:	2	4	16	3+1=4	F
STEP 5:	**2**	**4**	**16**	**4**	**F**

STEP 8: print **16**.

Step 1: Base 2
Step 2: Power 4
Step 3: Product Base
Step 4: Counter 1
Step 5: While Counter < Power
Repeat Step 5 through step 7
Step 6: Product Product * Base
Step 7: Counter Counter +1
Step 8: *Print* Product

- **Example 10:** Write down an algorithm and draw a flowchart to find and print the largest of three numbers. Read numbers one by one. Verify your result by a trace table. (Use 5, 7, 3 as the numbers read)

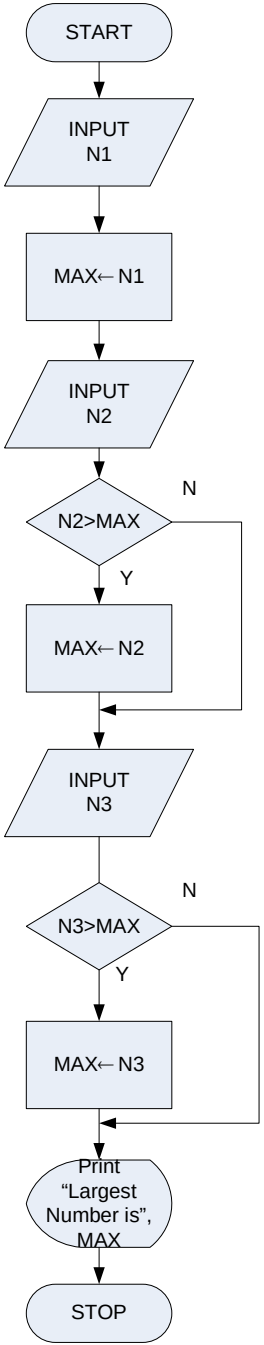
Algorithm

- Step 1: *Input* N1
- Step 2: Max N1
- Step 3: *Input* N2
- Step 4: *If (*N2>Max) *then*
  Max = N2
  endif
- Step 5: *Input* N3
- Step 6: *If (*N3>Max) then
  Max = N3
  endif
- Step 7: *Print* “The largest number is:”,Max

Flowchart & Tracing

	N1	N2	N3	Max	N2>Max	N3>Max
Step1:	5	?	?	?	?	?
Step 2:	5	?	?	5	?	?
Step 3:	5	7	?	5	T	?
Step 4:	5	7	?	7	T	?
Step 5:	5	7	3	7	F	F
Step 6:	5	7	3	7	F	F

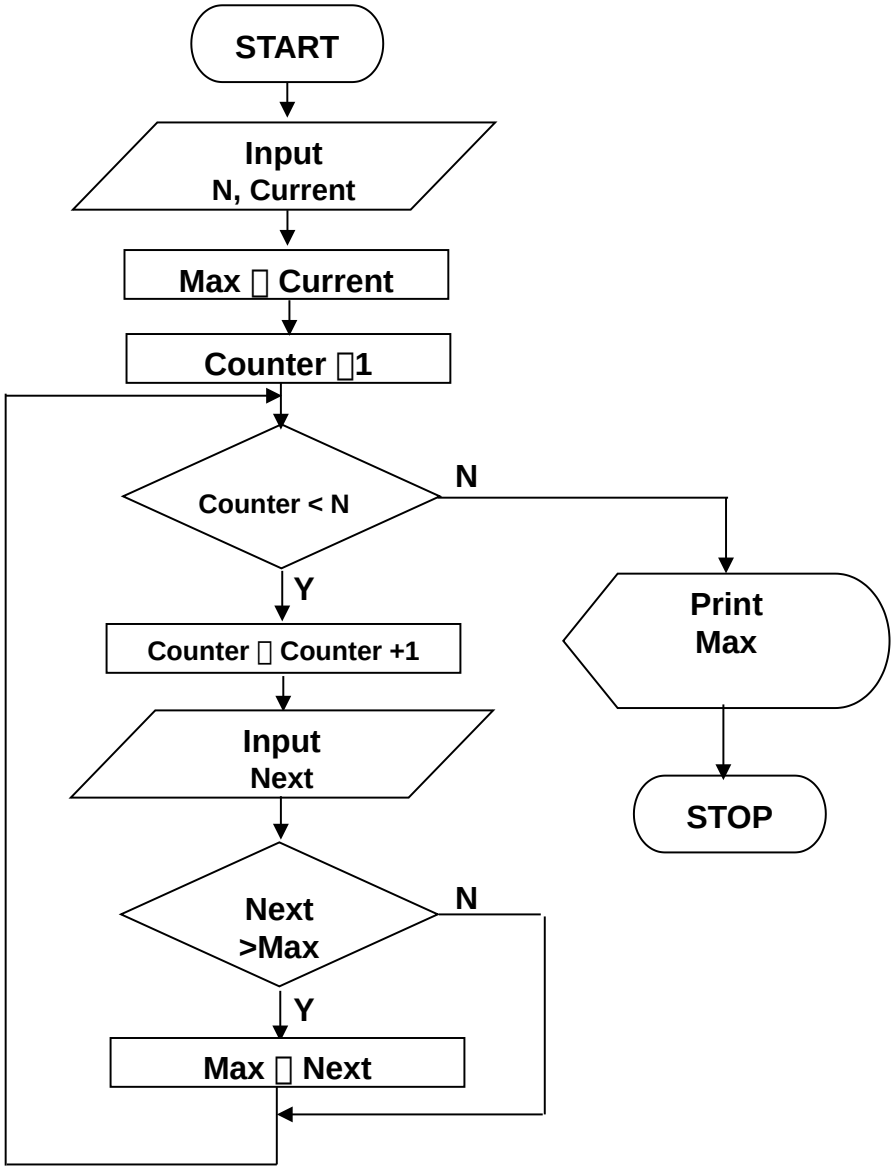
Step 8: Print Largest Number is **7**



- **Example 11**: Write down an algorithm and draw a flowchart to find and print the largest of N (N can be any number) numbers. Read numbers one by one. Verify your result by a trace table. (Assume N to be 5 and the following set to be the numbers {1 4 2 6 8 })

Algorithm:

- Step 1: *Input* N
- Step 2: *Input* Current
- Step 3: Max ← Current
- Step 4: Counter ←1
- Step 5: *While* (Counter < N)
  Repeat steps 5 through 8
- Step 6: Counter ← Counter + 1
- Step 7: *Input* Next
- Step 8: *If* (Next > Max) then
  Max ← Next
  endif
- Step 9: *Print* Max


START
Input
N, Current
Max Current
Counter 1
Counter < N
N
Y
Print
Max
Counter Counter +1
Input
Next
STOP
Next
>Max
N
Y
Max Next

Tracing

	N	Current	Max	Counter	Counter < N	Next	Next > Max
Step 1	5						
Step 2	5	1					
Step 3	5	1	1				
Step 4	5	1	1	1	T		
Step 5	5	1	1	1	T		
Step 6	5	1	1	2	T		
Step 7	5	1	1	2	T	4	
Step 8	5	1	4	2	T	4	T
Step 5	5	1	4	2	T	4	F
Step 6	5	1	4	3	T	4	F
Step 7	5	1	4	3	T	2	F
Step 8	5	1	4	3	T	2	F
Step 5	5	1	4	3	T	2	F
Step 6	5	1	4	4	T	2	F
Step 7	5	1	4	4	T	6	T
Step 8	5	1	6	4	T	6	T
Step 5	5	1	6	4	T	6	F
Step 6	5	1	6	5	F	6	F
Step 7	5	1	6	5	F	8	T
Step 8	5	1	8	5	F	8	T
Step 5	5	1	8	5	F	8	F
Step 9			**8 output**				

- **Prob. 1.** Write an algorithm and draw a flowchart to print the square of all numbers from LOW to HIGH. Test with LOW=1 and HIGH=10.
- **Prob. 2.** Write an algorithm and draw a flowchart to print the SUM of numbers from LOW to HIGH. Test with LOW=3 and HIGH=9.
- **Prob. 3.** Write an algorithm and draw a flowchart to print the SUM of numbers from LOW to HIGH. Test with LOW=3 and HIGH=9.
- **Prob. 4.** Write an algorithm and draw a flowchart to print all numbers between LOW and HIGH that are divisible by NUMBER.
- **Prob. 5.** Write an algorithm and draw a flowchart to print all the prime numbers between LOW and HIGH. Test with LOW=1 and HIGH=100.

- **Prob. 6.** Write an algorithm and draw a flowchart to count and print all numbers from LOW to HIGH by steps of STEP. Test with LOW=0 and HIGH=100.

- **Prob. 7.** Write an algorithm and draw a flowchart to count and print all numbers from HIGH to LOW by steps of STEP. Test with HIGH=100 and LOW=0.

- **Prob. 8.** Write an algorithm and draw a flowchart to print the multiplication table for 6's. i.e.
- ---- 1 6 = 6
- ---- 2 6 = 12

…

- ---- 12 6 = 72

- **Prob. 9.** Write an algorithm and draw a flowchart to print the complete multiplication table for 1's. through 12's.
- ---- 1 1 = 1, 1 2 = 2, … 1 12 = 12
- ---- 2 1 = 2, 2 2 = 4, … 2 12 = 24

  … …
- ---- 12 1 = 12, 12 2 = 24, …12 12 = 144

- **Prob. 10.** Write an algorithm and draw a flowchart to arrange N values read from the input in ascending order.
- **Prob. 11.** Write an algorithm and draw a flowchart that will find and print the product of 3 numbers.

- **Prob. 12.** Write an algorithm and draw a flowchart that will find and print
- The factorial of NUMBER is FACTORIAL.
- Test the flowchart for NUMBER=5.

- **Prob. 13.** Write an algorithm and draw a flowchart that will find and print the number of vowels in a given set of characters and print there number of occurrences.