

nth power of 2

$$\underline{\underline{n=4}}$$

$$\hookrightarrow 2^4 \text{ i.e., } 2^n \Rightarrow 16$$

$$2^2 = 1 \times 2 = 2$$

$\left\{ \begin{array}{l} \text{res} = 1; \\ \text{for (1... } n^{\text{th}} \text{ times)} \\ \{ \\ \quad \text{res} \times = 2; \\ \} \\ \text{Syso(res)} \end{array} \right.$

$$\begin{array}{l} n=0 \\ \textcircled{n=1} \\ n=2 \end{array}$$

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9
10        int ans = 1;
11
12        for(int i = 1 ; i <= n; i++){
13            ans = ans * 2;
14        }
15        System.out.println(ans);
16    }
17 }
```

$$\textcircled{\text{ans} = 1}$$

$$\begin{array}{l} n=1 \\ 2^1 = 2 \end{array}$$

$$\begin{array}{l} n=2 \\ \textcircled{2^2} = 4 \\ \boxed{2 \times 2} \end{array}$$

$$\begin{array}{l} \checkmark \\ \textcircled{n=0} \rightarrow 1 \\ 2^0 \rightarrow 1 \end{array}$$

$$\begin{array}{l} n=3 \\ 2^3 = 8 \\ \boxed{2 \times 2 \times 2} \end{array}$$

Print powers of 2 less than n

res =  $2^0$  1  
 $2^1$  2  
 $2^2$  4  
 $2^3$  8  
 $2^4$  16  
 $2^5$  32  
 $2^6$  64  
 $2^7$  128  
 $2^8$  256

i/p n=50 ✓  
 1  
 2  
 4  
 8  
 16  
 32  
 64  
 128  
 256

n=70  
 1  
 2  
 4  
 8  
 16  
 32  
 64

n=100

1. Hint

start with  $\rightarrow 1$   
 $i = 1$   
 $i = 1$

for (int i=1;  $i < n$ ;  $i *= 2$ )  
 {  
 syso ( i );  
 }

$2^0$  1  
 $2^1$  2  
 $2^2$  4  
 8  
 16  
 32

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        for(int i = 1; i < n; i *= 2){
            System.out.println(i);
        }
    }
}
```

n=50

$i = 1$  2 4 8 16 32 64

1 < 50  
 2 < 50  
 4 < 50  
 8 < 50  
 16 < 50  
 32 < 50

64 < 50  
 ✗

$$2^n$$



```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9
10        int ans = 1;
11
12        for(int i = 1 ; i <= n; i++){
13            ans = ans * 2;
14        }
15        System.out.println(ans);
16    }
17 }

```

```

import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        for(int i = 1; i < n; i *= 2){
            System.out.println(i);
        }
    }
}

```

$$2^n$$

$$n=0, 1, 2, 4, 8, \dots$$

$$2^1 \dots 2^n$$

using loop  $\rightarrow$  computing

$$n=0$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9
10        int ans = 1;
11
12        for(int i = 1 ; i <= n; i++){
13            ans = ans * 2;
14        }
15        System.out.println(ans);
16    }
17 }

```

$$n=4 \rightarrow 2^4 = 16$$

$$ans=1$$

$$1 \times 2 \times 2 \times 2 \times 2 =$$

Print

n/3.

~~n = 80~~

~~26/3~~

~~8~~

70

28

80  
26

26  
8  
2

3  $\sqrt{8}$  2

6  
2

3  $\sqrt{80}$

6  
20  
18

2

3  $\sqrt{8}$   
26  
24  
2

70

2/3 = 0

0

i = 1

i < n ;

i % 2

i = n

i > 0

i % 3

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        for(int i = n; i > 0; i /= 3){
            System.out.println(i);
        }
    }
}
```

GKSTR12 Multiples of 3, 5 and Both 3 and 5

n = 16 ✓

n = 15

[1 ... n]

3 5 6 9 10 12 15

15  
/ 3 = 5

for ( [ 1 → n ] )  
{  
    (i % 3 == 0) or  
    syso(i)  
}

(i % 5 == 0)

3 | 15 = 5  
    15  
    —  
    0

5 | 10 = 2  
    10  
    —  
    0

Any ① → True.

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        for(int i = 1; i <= n; i++){
            if((i % 3 == 0) || (i % 5 == 0)){
                System.out.print(i + " ");
            }
        }
    }
}
```

15

✓  
15 % 3 == 0

# Running Sum for loop

Sample Input 0



Sample Output 0



sum = 0

for (                      x 5 )

{

sum = ~~7~~ 10

i/p  $\Rightarrow$  ① ② ③ ④ ⑤  
③ // ② // ① // ④

sum += i/p

sys.o (sum);

}



④  $\rightarrow$  n  
8 - ✓  
0 - ✓  
-2 - ✓  
3 - ✓  
sum = 8 + 2  
9

sum = 0

for (                      x 4 )

i/p  $\Rightarrow$  8 / 0 // -2 // ③

sum += i/p

sys.o (sum)

8  
8  
6  
9

③  $\rightarrow$  n

sum = 0

for (                      x 3 )

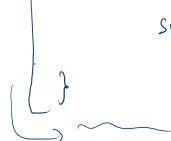
{

i/p  $\Rightarrow$  6 / 0 / -4

sum += i/p

sys.o (sum)

}



6  
0  
-4  
sum = 6 / 6 / 2

6  
6  
2

```

import java.io.*;
import java.util.*;

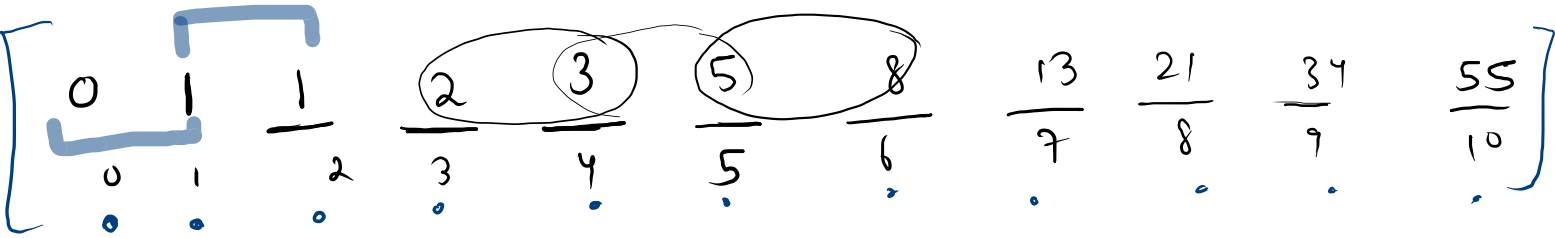
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

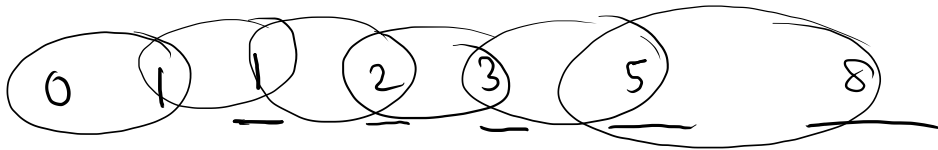
        int sum = 0;

        for(int i = 1; i <= n; i++){
            //
            take ip
            modify sum
            print sum
            //
            int val = scn.nextInt();
            sum = sum + val;
            System.out.println(sum);
        }
    }
}
    
```

# Fibonacci Series.



$$n^{\text{th}} = (n-1)^{\text{th}} + (n-2)^{\text{th}}$$



13 . . . .

Fibonacci number 12

.