nth power of 2

Problem | Submissions | Leaderboard | Discussions

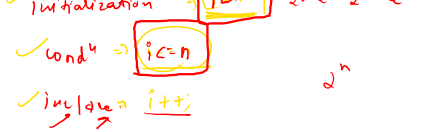
Take n as an integer input and you have to print the nth power of 2 as an integer output.

Input Format

n=6

o/p → 64

$2^6 = 64$

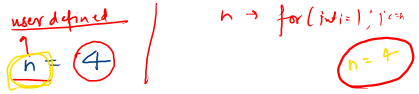


initialization ⇒ i=1

cond^n ⇒ i<=n

inc/dec ⇒ i++;

2^n

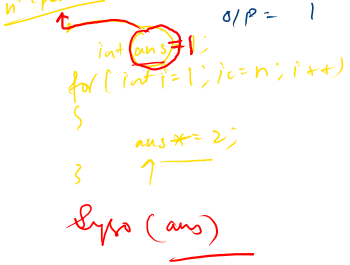


o/p = 2^4 = 16

n = 0

o/p = 1

$2^0 = 1$



ans=1;

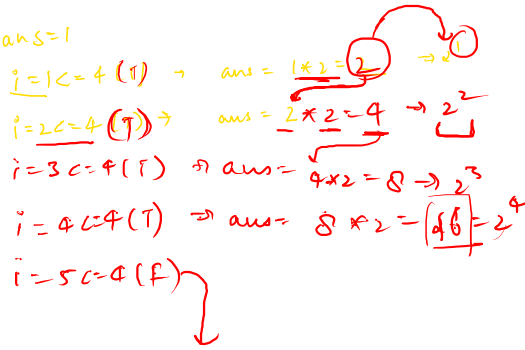
for(int i=1; i<=n; i++)

{

ans *= 2;

}

Syso(ans)



ans=1

i=1 <= 4 (T) → ans = 1*2 = 2 → 2^1

i=2 <= 4 (T) → ans = 2*2 = 4 → 2^2

i=3 <= 4 (T) → ans = 4*2 = 8 → 2^3

i=4 <= 4 (T) → ans = 8*2 = 16 → 2^4

i=5 <= 4 (F)

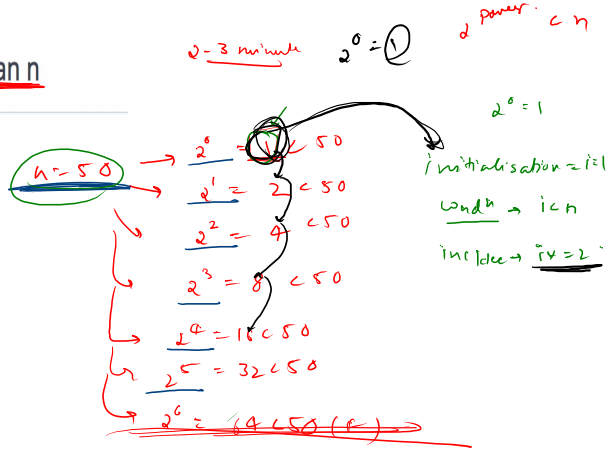
Print powers of 2 less than n

Sample Input 0

n = 50

Sample Output 0

1
2
4
8
16
32




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

i = 1 < 50 (T) → 1
i = 2 < 50 (T) → 2
i = 4 < 50 (T) → 4
i = 8 < 50 (T) → 8
i = 16 < 50 (T) → 16
i = 32 < 50 (T) → 32
i = 64 < 50 (F)

Print n/3

Problem | Submissions | Leaderboard | Discussions

Take n as an integer input from the user. Keep on dividing n by 3 and print the resultant value of n each time in a separate line, till the value of n is greater than 0.

Note: Start printing from n, keep on updating n by dividing n by 3 each time, and print the the updated value of n each time.

Sample Input 0

```
80
```

Sample Output 0

```
80
26
8
2
```

n = 80

$\frac{80}{3} = 26$ ....

$\frac{26}{3}$

80
26
8
2

initialization → i = n;

Cond^{n} → i > 0

inc/dec → i /= 3

Running Sum for loop

Problem | Submissions | Leaderboard | Discussions

You will be given a number n of integer data-type.

After this you will be given n integers as input of integer data-type. And you have to print the sum after you take input of an integer each time.

Initially the sum is zero.

Sample Input 0

```
5
3
2
2
-1
4
```

Sample Output 0

```
3
5
7
6
10
```

```
n = 5 , int sum = 0.
for(int i=1; i<=n; i++)
{
    int x = scn.nextInt();
    sum += x;
    sysos(sum);
}
```

OIP

3
5
7
6
10

Sum = 0 + 2

sum = 3

sum = 3 + 2 = 5

sum = 5 + 2 = 7

sum = 7 - 1 = 6

sum = 6 + 4 = 10

→ i = 1 c = 5 (T)

→ i = 2 c = 5 (T)

→ i = 3 c = 5 (T)

→ i = 4 c = 5 (T)

→ i = 5 c = 5 (T)

→ i = 6 c = 5 (F)

```
0
1       int n=scn.nextInt();
2       int sum=0;
3
4 //        for(int i=1;i<=n;i++){
5 //            int x=scn.nextInt();
6 //            sum+=x;
7 //            System.out.println(sum);
8
9 //        }
```

sum = 0;

O/P

n = 5

3
2
2
-1
4

i=1<=5(T) → x = 3 — sum = 0+3 = 3 — 3

i=2<=5(T) → x = 2 — sum = 3+2 = 5 — 5

i=3<=5(T) → x = 2 — sum = 5+2 = 7 — 7

i=4<=5(T) → x = -1 — sum = 7-1 = 6 — 6

i=5<=5(T) → x = 4 — sum = 6+4 = 10 — 10

i=6<=5(F)