



Utopian Tree ★

9.8 more points to get your gold badge!

Rank: 265048 | Points: 840.2/850

**Your Utopian Tree submission got 20.00 points.**[Share](#)[Post](#)

You are now 9.8 points away from the gold level for your problem solving badge.

[Try the next challenge](#) | [Try a Random Challenge](#)[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Topics](#)

The Utopian Tree goes through 2 cycles of growth every year. Each spring, it doubles in height. Each summer, its height increases by 1 meter.

A Utopian Tree sapling with a height of 1 meter is planted at the onset of spring. How tall will the tree be after n growth cycles?

For example, if the number of growth cycles is $n = 5$, the calculations are as follows:

Period	Height
0	1
1	2
2	3
3	6
4	7
5	14

Function Description

Complete the `utopianTree` function in the editor below.

`utopianTree` has the following parameter(s):

- `int n`: the number of growth cycles to simulate

Returns

- `int`: the height of the tree after the given number of cycles

Input Format

The first line contains an integer, t , the number of test cases.

t subsequent lines each contain an integer, n , the number of cycles for that test case.

Constraints

$$1 \leq t \leq 10$$

$$0 \leq n \leq 60$$

Sample Input

```
3
0
1
4
```

Sample Output

```
1
2
7
```

Explanation

There are 3 test cases.

In the first case ($n = 0$), the initial height ($H = 1$) of the tree remains unchanged.

In the second case ($n = 1$), the tree doubles in height and is **2** meters tall after the spring cycle.

In the third case ($n = 4$), the tree doubles its height in spring ($n = 1, H = 2$), then grows a meter in summer ($n = 2, H = 3$), then doubles after the next spring ($n = 3, H = 6$), and grows another meter after summer ($n = 4, H = 7$). Thus, at the end of 4 cycles, its height is **7** meters.

[Change Theme](#)

Language

Python 3



```

1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9  #
10 # Complete the 'utopianTree' function below.
11 #
12 # The function is expected to return an INTEGER.
13 # The function accepts INTEGER n as parameter.
14 #
15
16 def utopianTree(n):
17     # Write your code here
18     return 2 ** (math.floor((n + 1) / 2) + 1) - 1 - (n % 2)
19
20
21 if __name__ == '__main__':
22     fptr = open(os.environ['OUTPUT_PATH'], 'w')
23
24     t = int(input().strip())
25
26     for t_itr in range(t):
27         n = int(input().strip())
28
29         result = utopianTree(n)
30
31         fptr.write(str(result) + '\n')
32
33     fptr.close()
34

```

EMACS

Line: 20 Col: 1

Upload Code as File



Test against custom input

Run Code

Submit Code

You have earned 20.00 points!

You are now 9.8 points away from the gold level for your problem solving badge.

97%

840.2/850



Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Compiler Message

Test case 1

Success

Test case 2

Input (stdin)

Download

1 2
2 0
3 1

Test case 3

Test case 4

Expected Output

Download

1 1
2 2

Test case 5

Test case 6