#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

// Complete the utopianTree function below.

int utopianTree(int n) {

if(n==0)

{

return 1;

}

else {

int i;

int height;

height = 1;

for(i=0;i<n;i++)

{

if(i%2==0)

{

height = height \* 2;

}

else {

height++;

}

}

return height;

}

}

int main()

{

FILE\* fptr = fopen(getenv("OUTPUT\_PATH"), "w");

char\* t\_endptr;

char\* t\_str = readline();

int t = strtol(t\_str, &t\_endptr, 10);

if (t\_endptr == t\_str || \*t\_endptr != '\0') { exit(EXIT\_FAILURE); }

for (int t\_itr = 0; t\_itr < t; t\_itr++) {

char\* n\_endptr;

char\* n\_str = readline();

int n = strtol(n\_str, &n\_endptr, 10);

if (n\_endptr == n\_str || \*n\_endptr != '\0') { exit(EXIT\_FAILURE); }

int result = utopianTree(n);

fprintf(fptr, "%d\n", result);

}

fclose(fptr);

return 0;

}

char\* readline() {

size\_t alloc\_length = 1024;

size\_t data\_length = 0;

char\* data = malloc(alloc\_length);

while (true) {

char\* cursor = data + data\_length;

char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

if (!line) { break; }

data\_length += strlen(cursor);

if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') { break; }

size\_t new\_length = alloc\_length << 1;

data = realloc(data, new\_length);

if (!data) { break; }

alloc\_length = new\_length;

}

if (data[data\_length - 1] == '\n') {

data[data\_length - 1] = '\0';

}

data = realloc(data, data\_length);

return data;

}