Crossroads
The ACM Student Magazine

Computational Recreations: The *n* Days of Christmas

by ***Jonathan Doyle***

In what will likely become a regular feature, this issue of *Crossroads* introduces the Computational Recreations column. These columns will include puzzles, games, and brainteasers intended to challenge and tickle the mind.

The *n* Days of Christmas

In the well-known song *The Twelve Days of Christmas*, each verse builds upon the previous one to describe an ever-increasing influx of gifts. But why stop after a mere twelve days when we can extend the song? Instead of a fixed number, we can continue for some arbitrary number of days!

This new song, *The* n *Days of Christmas*, continues the pattern established in the original version:

1. On the first day, you receive a present of type T_1.
2. On the second day, you receive two presents of type T_2, plus one of type T_1.
3. On the third day, you receive three presents of type T_3, two of type T_2, and one of type T_1.

   ...
4. On the n^{th} day, you receive n presents of type T_n, plus $n-1$ of type T_{n-1}, $n-2$ of type T_{n-2}, and so forth, down to one present of type T_1.

Questions

1. How many presents do you receive on day n?

2. Assuming you keep all presents, how many will have accumulated in total by day *n*?
   (Hint: 78 gifts are received on day 12, bringing the total to 364.)
3. Let $S_1(n)$ be the answer to question 1 and $S_2(n)$ be the answer to question 2. It is clear that $S_2(n)$ will be a sum of sums:

   $$S_2(n) = S_1(1) + S_1(2) + S_1(3) + \ldots + S_1(n)$$

   We can continue this pattern and define $S_k(n)$ as follows:

   $$S_k(n) = S_{k-1}(1) + S_{k-1}(2) + S_{k-1}(3) + \ldots + S_{k-1}(n)$$

   Is there a non-recursive function to describe $S_k(n)$?
4. You wish to implement the function you calculated above in the C language. Unfortunately, the obvious way to do it will result in integer overflow for relatively small values of *n*. This overflow occurs in numbers that are used for intermediate calculations; the final result is much smaller. Can the function be written to avoid using numbers that do not fit into an `int` for all cases where the final output also fits into an `int`?

References

This problem was thought to be original until the first Google search for the title showed otherwise. There does not appear to be a clear indication of who first posed the problem.

Biography

Jonathan Doyle is a graduate student at Dalhousie University. He believes every day is Christmas.