

Climb ladder

A person is standing at the bottom of a ladder that has $N > 0$ rungs.

The person can climb the ladder by advancing each step 1 rung at a time, 2 rungs at a time, or 3 rungs at a time.



How many different ways are there for someone starting from the ground at the ladder's base to climb the ladder such that their feet end on the ladder's top rung?

Write a function

```
int nclimbs(int n)
```

where

n is the number of rungs on the ladder

and returns the number of ways to climb the ladder if $n > 0$, otherwise returns -1

File you must submit: `soln_func.cc`

Examples:

$n=1$

Returns: 1

Explanation: Only one way, climb a single rung.

$n=2$

Returns: 2

Explanation: 1+1 or 2 rungs in a single step.

$n=4$

Returns: 7

Explanation: The seven different ways are:

$1+1+1+1$

$1+1+2$

$1+2+1$

$2+1+1$

$2+2$

$3+1$

$1+3$

$n=0$

$n=-1$

Each returns: -1

Explanation: Neither satisfies $n > 0$.