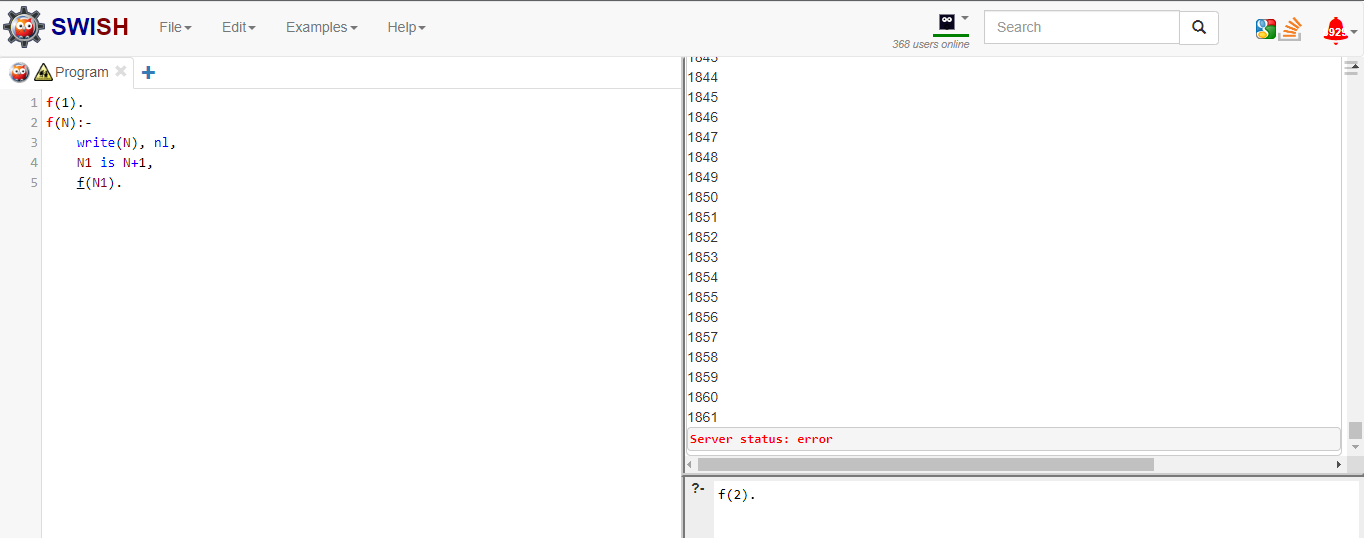
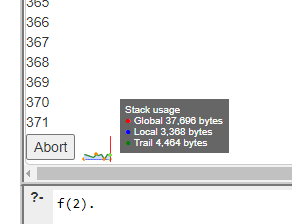
Attempt to cause Stack Overflow in recursive programs

*Objective:*

* Cause stack overflow error in *Prolog*, *C++* (gcc below 7.0 and gcc 7.0 or above) and *Python*

1. Online Swish Prolog attempt:

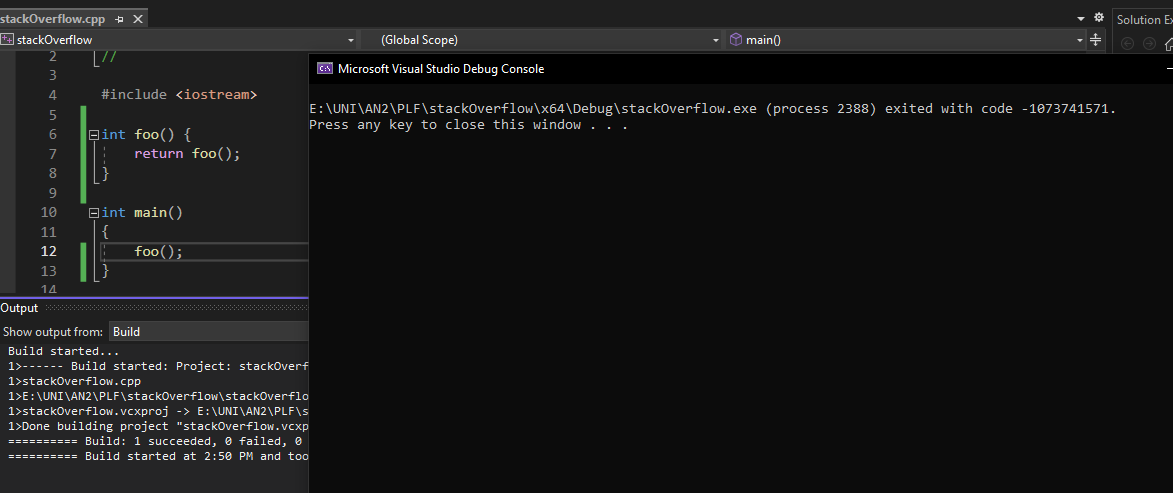
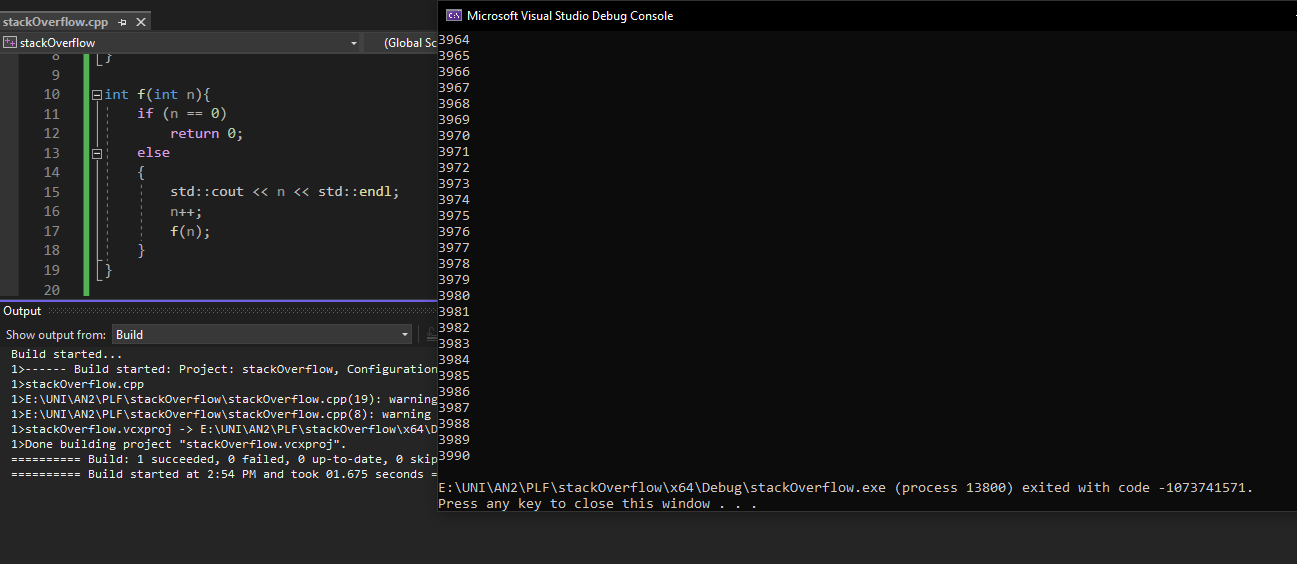


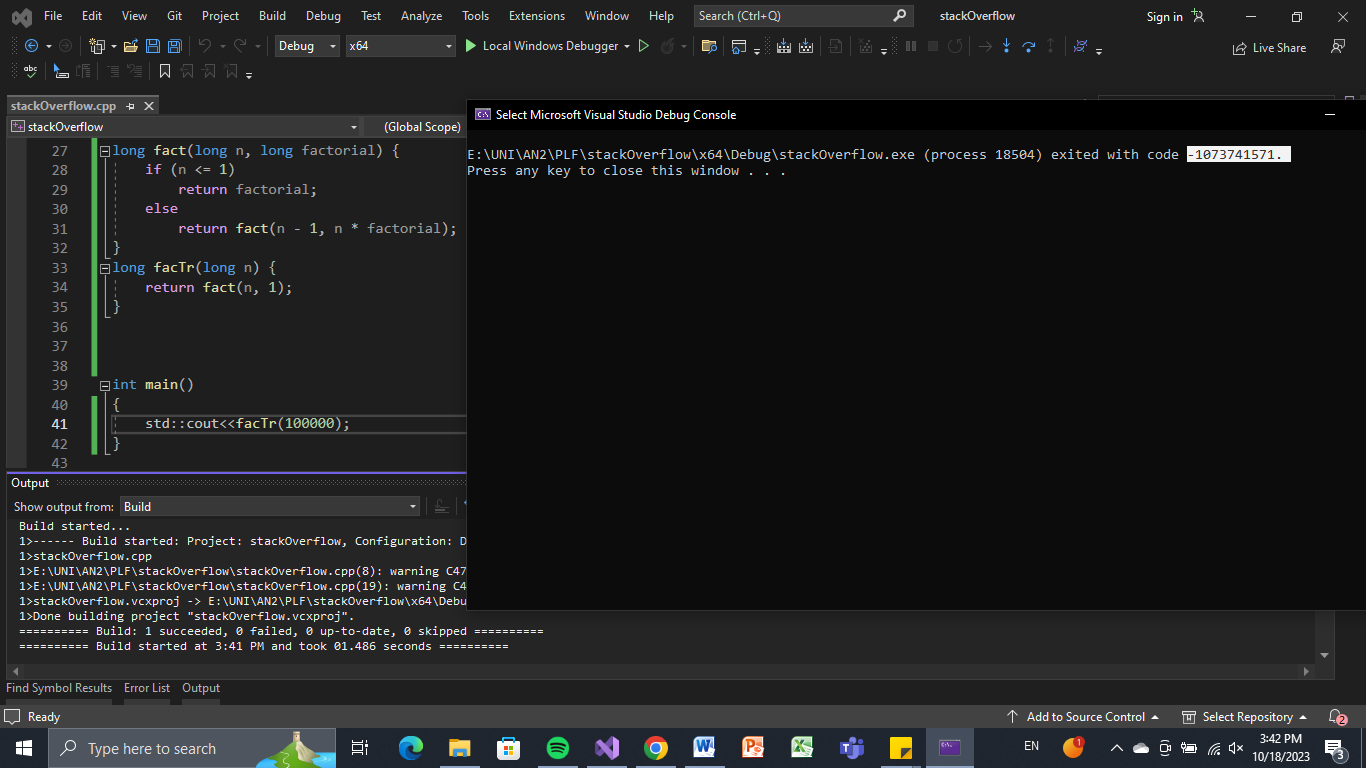
*Conclusion*: “Server status: error” after about 1900 steps

1. Prolog attempt

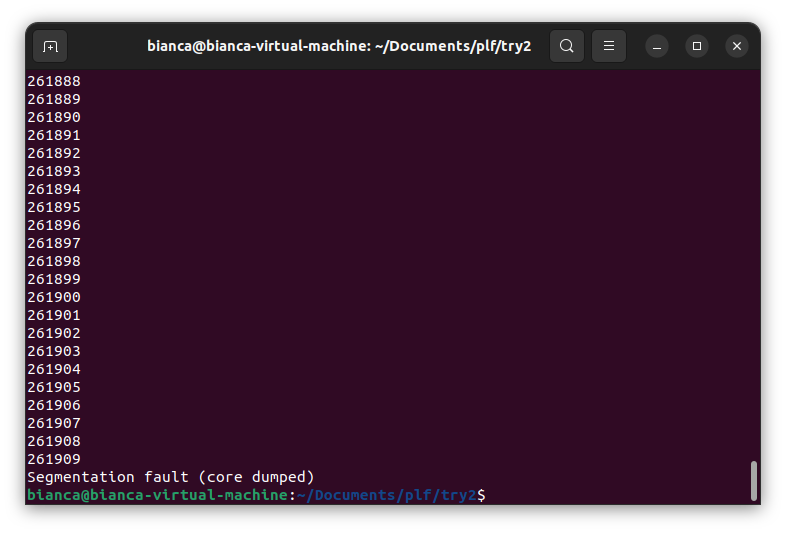
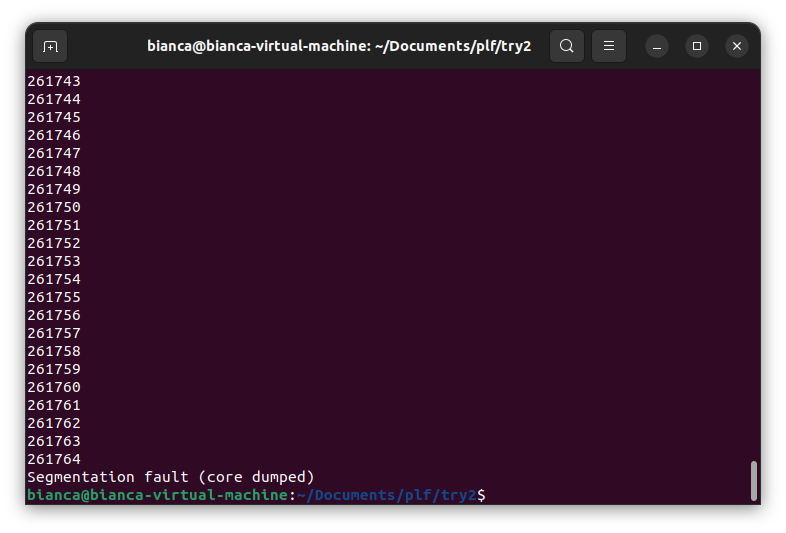
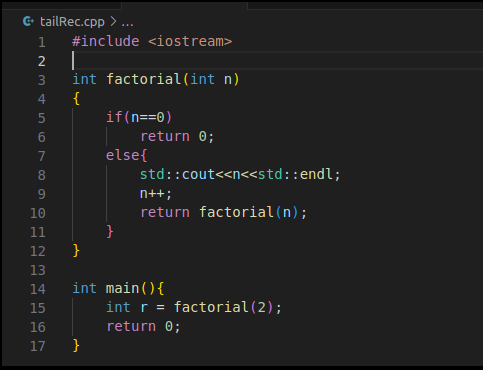
*Conclusion:* Stack Overflow *wasn’t triggered*; the window minimized and stopped responding, so I was unable to tell how many steps it got to (>10k). Significantly faster than the online version, but ultimately I had to end task.

1. C++ gcc 6.3.0

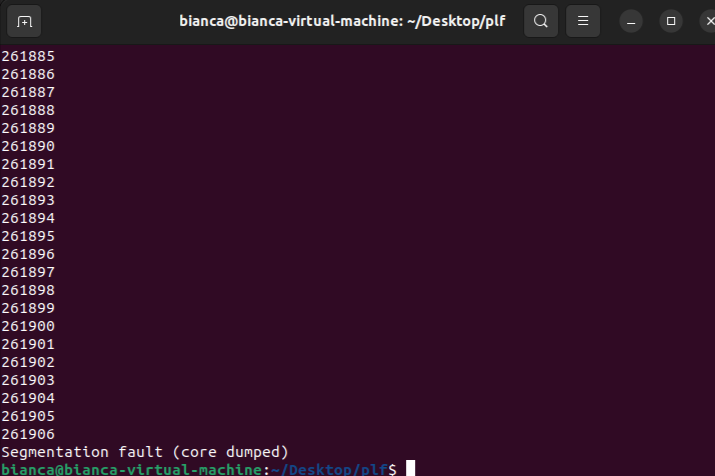
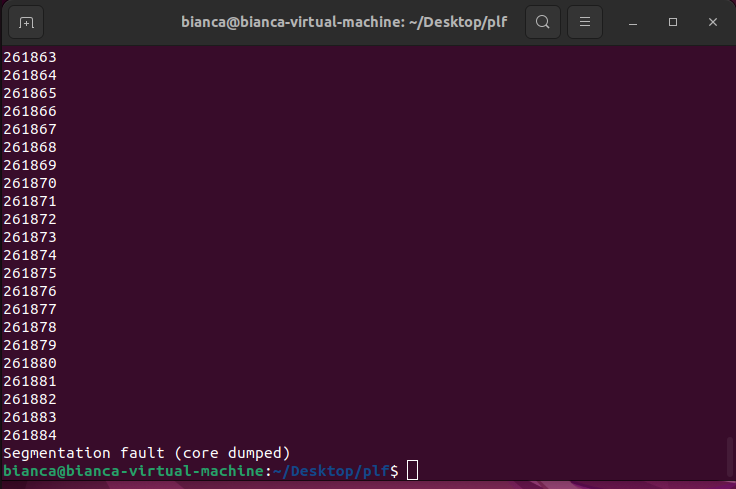
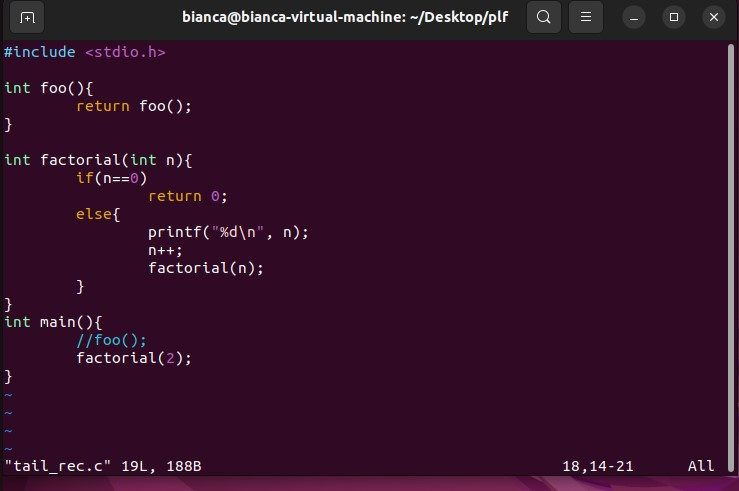




*Conclusion:* Stack Overflow after about 4k steps (exit code 0xC00000FD)

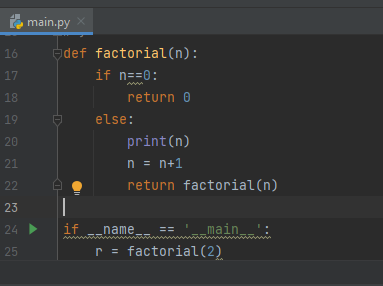
1. C++ gcc 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)

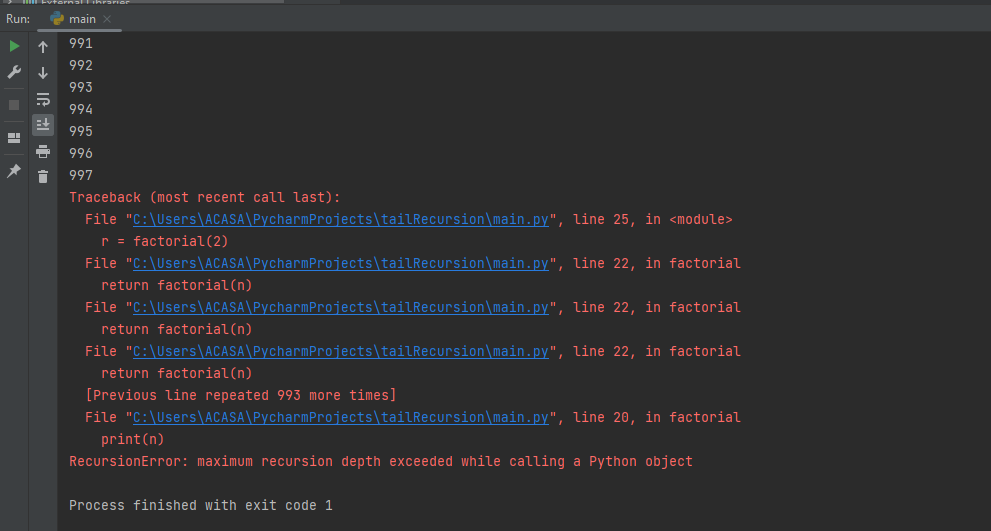
*Conclusion*: After around 26k steps, a segmentation fault occurs (caused by the overflowing of the stack)

1. Using C:

*Conclusion*: Stack Overflow occurs after a variable number of steps

1. Python:

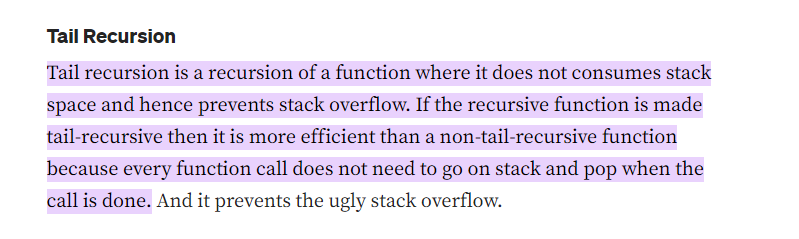




*Conclusion:* After about 1k steps, maximum recursion depth is exceeded (Stack Overflow)

**Final conclusions:**

* Stack Overflow was caused using tail recursion on both gcc below 7.0 and above, in Python and in C/C++
* Prolog was the only one that behaved differently (inconclusive); further testing on different machines with varying levels of performance is needed.

*References:*

(Source: <https://vijeshsalian.medium.com/recursion-how-to-overflow-the-stack-and-how-not-to-b9dcffdfab27#:~:text=Tail%20recursion%20is%20a%20recursion,when%20the%20call%20is%20done>.)