Simulating recursion (1) P62390_en

In C++, consider this program (whose inclusions have been removed):

```
void work(int n) {
    if (n > 0) {
        cout << ' ' << n;
        work(n − 1);
        work(n − 1);
    }
}

int main() {
    int n;
    while (cin >> n) {
        work(n);
        cout << endl;
    }
}
```

In Python, consider this program:

```
from yogi import tokens

def work(n: int) −> None:
    if n > 0:
        print(' ', n, end='')
        work(n − 1)
        work(n − 1)

def main() −> None:
    for n in tokens(int):
        work(n)
        print()

main()
```

Take a look at the sample input and sample output to see what this program prints for every given number.

Without modifying @main()@, reimplement the procedure @work(n)@ with no calls at all, recursive or not, so that the output of the program does not change.

Input

Input consists of several strictly positive natural numbers.

Output

For every number, print a line identical to the one written by the program above.

Observation

To solve this exercise, the only containers that you should use are stacks.

Sample input 1

```
1
2
3
4
```

Sample output 1

```
1
2 1 1
3 2 1 1 2 1 1
4 3 2 1 1 2 1 1 3 2 1 1 2 1 1
```

Problem information

Author: Salvador Roura
Translator: Carlos Molina

Generation: 2026-01-25T11:13:48.331Z