

---

## Simulant recursivitat (1)

P62390\_ca

---

EN C++, considereu aquest programa (del qual s'han suprimit les incusions):

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

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

En Python, considereu aquest programa:

```
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()
```

A l'exemple d'entrada i de sortida d'aquest exercici veureu què escriu aquest programa per a cada nombre llegit.

Sense modificar gens el *main()*, reimplementeu el procediment *escriu (n)* sense fer cap crida, recursiva o no, de manera que la sortida del programa no canviï.

### Entrada

L'entrada consisteix en diversos naturals estrictament positius.

### Sortida

Per a cada nombre, escriviu una línia idèntica a l'escripta pel programa donat.

## Observació

Per resoldre aquest exercici, els únics contenidors que hauríeu d'usar són piles.

### Exemple d'entrada

```
1
2
3
4
```

### Exemple de sortida

```
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
```

## Informació del problema

Autor : Salvador Roura

Generació : 2024-05-02 20:47:59

© *Jutge.org*, 2006–2024.

<https://jutge.org>