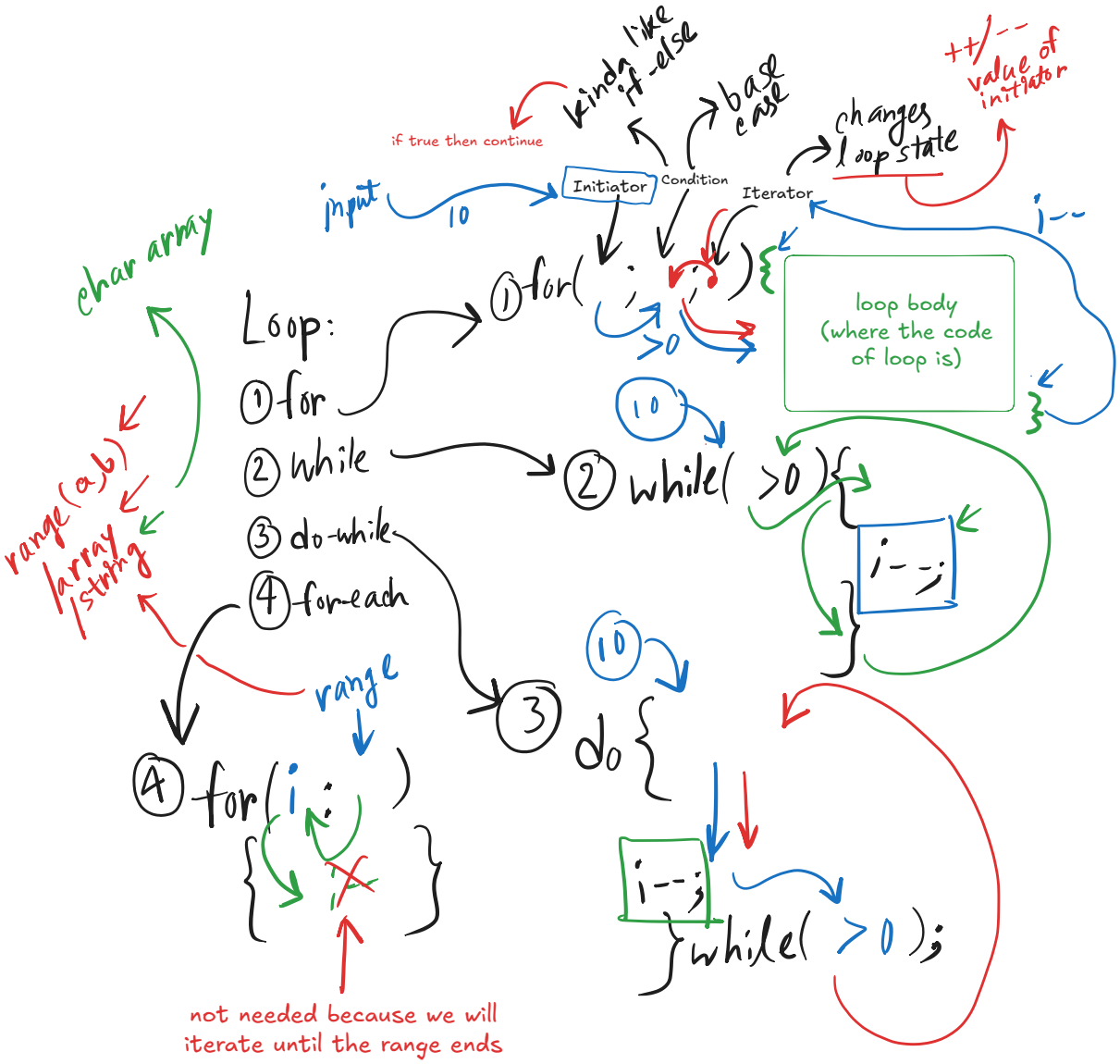

C++ does not have a built-in `range()` function exactly like Python's, but **the C++20 Ranges library provides an elegant and lazy-evaluated equivalent using** `std::views::iota`. For older C++ versions, a traditional `for` loop or a custom class is needed.

C++20 Solution (`std::views::iota`)

The most "Pythonic" and efficient way in modern C++ (C++20 and later) is to use `std::views::iota`. This generates a sequence of numbers on demand (lazy evaluation), just like Python's `range()` function.

cpp

```cpp
#include <algorithm>
#include <ranges>
#include <iostream>

int main() {
    // Python equivalent: range(10) -> 0 1 2 3 4 5 6 7 8 9
    for (int i : std::views::iota(0, 10)) {
        std::cout << i << " ";
    }
    std::cout << std::endl;

    // Python equivalent: range(4, 10) -> 4 5 6 7 8 9
    for (int i : std::views::iota(4, 10)) {
        std::cout << i << " ";
    }
    std::cout << std::endl;
}
```