



Iterator invalidation

```
auto numbers = std::vector { 1, 2, 3 };  
auto iterator = numbers.begin();  
numbers = std::vector { 6, 7, 8 };  
  
while (iterator != numbers.end())  
    std::print("{}\n", *iterator++);
```

```
-std=c++23 -Wall -Wextra -Wpedantic  
  
ASM generation compiler returned: 0  
Execution build compiler returned: 0  
Program returned: 0  
1672498001  
5  
1500785022  
1062499325  
0  
0  
33  
0  
6  
7  
8
```



Rust (unsafe)

```
let mut numbers = vec![1, 2, 3];
let mut iterator = numbers.as_ptr();
numbers = vec![6, 7, 8];

// SAFETY: This is unsafe and could lead to undefined behavior
unsafe {
    while iterator < numbers.as_ptr().add(numbers.len()) {
        println!("{}", *iterator);
        iterator = iterator.add(1);
    }
}
```

```
Program returned: 0
286842394
6
-1215374705
1641395135
0
0
33
0
6
7
8
```