The source code originates from https://marcuscode.com/2020/10/finding-factorial-in-ccpp

```
#include <stdio.h>

int main() {
    int n, i;
    unsigned long long factorial = 1;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    for (i = 1; i <= n; ++i) {
        factorial *= i;
    }

    printf("Factorial of %d = %llu", n, factorial);
    return 0;
}
```

Input Validation:

- Problem: The program does not validate if the user inputs a positive integer. If a negative number or a non-integer is entered, the behavior is undefined.
- Solution: Implement checks to ensure the input is a non-negative integer.

Integer Overflow:

- Problem: The variable `factorial` is of type `unsigned long long`, which may still overflow for large values of n (e.g., `n > 20`).
- Solution: While `unsigned long long` can handle larger values, it's still limited. For extremely large n, consider using arbitrary-precision arithmetic libraries.

The program after improvement.

```
#include <stdio.h>

int main() {
    int n, i;
    unsigned long long factorial = 1;

    printf("Enter a non-negative integer: ");
    if (scanf("%d", &n) != 1) {
        printf("Invalid input! Please enter an integer.\n");
```

```
        return 1;
    }

    if (n < 0) {
        printf("Factorial is not defined for negative numbers.\n");
        return 1;
    }

    for (i = 1; i <= n; ++i) {
        factorial *= i;
    }

    printf("Factorial of %d = %llu\n", n, factorial);
    return 0;
}
```