```
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {
    threadRunning = true;
    while (pow (x, n) + pow (y, n) != pow (z, n)) ++n;

    return false;

}

void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```

1. A function that never returns is UB: C++ may assume that functions will always return*

2. **The only way this function can return is be returning false**

3. The function does not have any side-effects

* if it doesn't call any IO or `[[no_return]]` is not specified

```cpp
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {
    threadRunning = true;
    while (pow (x, n) + pow (y, n) != pow (z, n)) ++n;

    return false;

}

void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```

1. **A function that never returns is UB: C++ may assume that functions will always return***
2. **The only way this function can return is be returning false**
3. **The function does not have any side-effects**

*** if it doesn't call any IO or `[[no_return]]` is not specified**

```cpp
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {


    return false;

}

void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```