**Review Questions and Exercises**

**10.25** How do smart pointers differ from regular pointers?

**Smart pointers can automatically deallocate resources once they are out of the scope of use which prevents the problem of dangling pointers and memory leaks which is possible with regular pointers if not deleted.**

**10.26** Name the header file that needs to be included in a program that uses smart pointers.

**#include <memory>**

**10.27** What happens when a unique\_ptr that is managing an object is assigned the nullptr value?

**The managed object is deallocated.**

**10.28** What does the get() method of the unique\_ptr class do?

**get() returns the raw pointer to the object managed by this smart pointer. This is useful if such a pointer needs to be passed to a function that does not know how to handle smart pointers.**

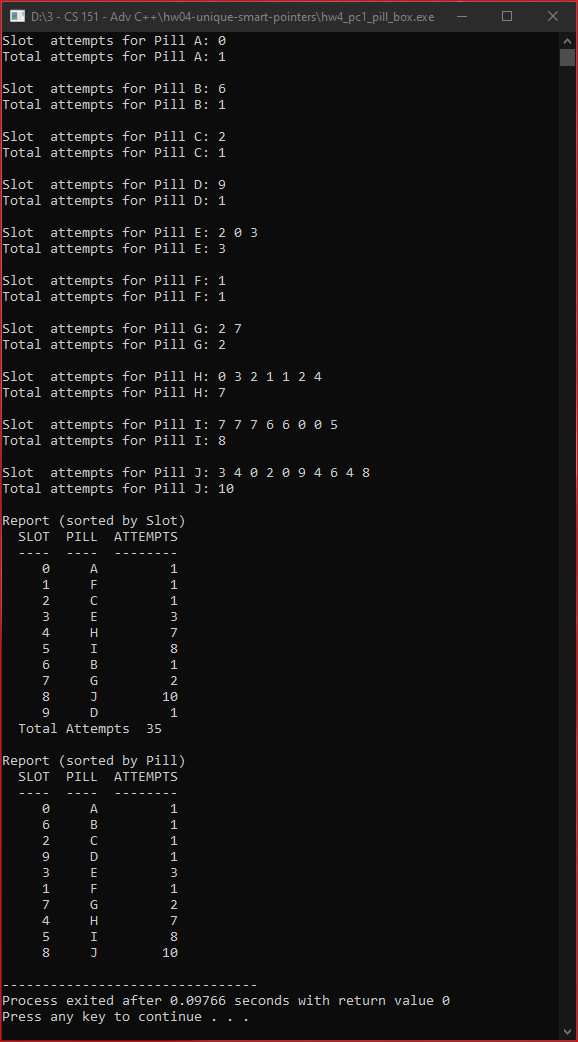
**10.30** List three different operations that are permitted on raw pointers but not on unique\_ptr objects.

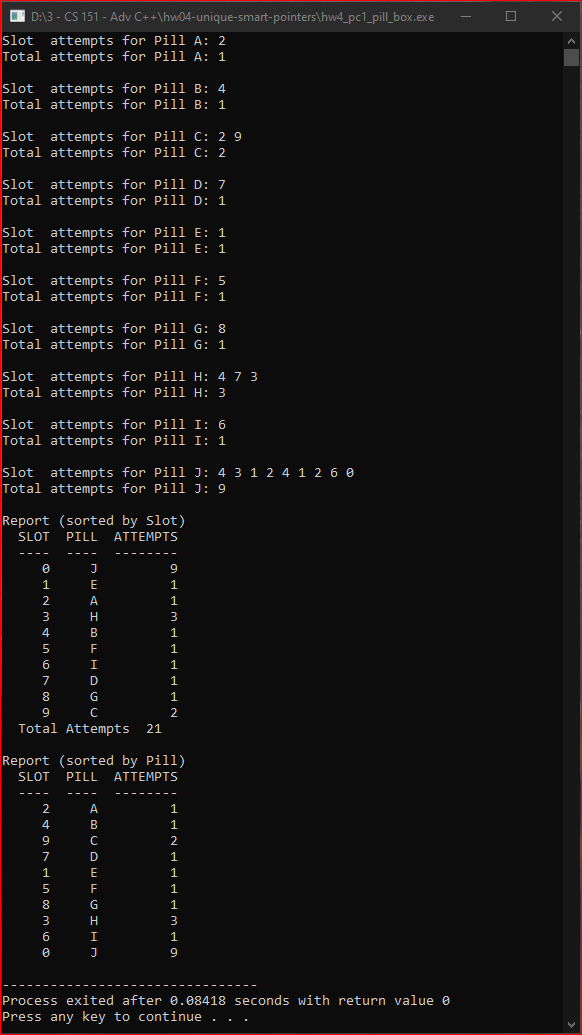
**1. Arithmetic operations e.g. uptr++, uptr— is not valid**

**2. Initialize value of another unique\_ptr e.g. unique\_ptr<int> uptr2 = uptr1 is not valid**

**3. Directly pass pointer to a function by value because pass by value means the argument’s value is copied into the value of the corresponding function parameter**

**Screenshots**

****

****