CS 300 Data Structures
Problem Set # 13 – C++ Smart Pointers

1. Write a program that defines a unique pointer to an integer value. Use the `std::make_unique` function to create a pointer.

```
#include <iostream>
#include <memory>

int main()
{
    std::unique_ptr<int> p = std::make_unique<int>(123);
    std::cout << "The value of a pointed-to object is: " << *p << '\n';
}
```

2. Write a program that defines a class with two data members, a user-defined constructor, and one member function. Create a unique pointer to an object of a class. Use the smart pointer to access the member function.

```
#include <iostream>
#include <memory>

class MyClass
{
   private:
      int x;
      double d;
   public:
      MyClass(int xx, double dd)
          : x{ xx }, d{ dd }
      {}
      void print()
      {
         std::cout << x << " and: " << d;
      }
};
int main()
{
   std::unique_ptr<MyClass> p = std::make_unique<MyClass>(123, 456.789);
   p->print();
}
```

3. Write a program that defines three shared pointers pointing at the same object of type int. Create the first pointer through an `std::make_shared` function. Create the remaining pointers by copying the first pointer. Access the pointed-to object through all the pointers.

```
#include <iostream>
#include <memory>

int main()
{
    std::shared_ptr<int> p1 = std::make_shared<int>(123);
    std::shared_ptr<int> p2 = p1;
    std::shared_ptr<int> p3 = p1;

    std::cout << "Value accessed through a first pointer: " << *p1 << '\n';
    std::cout << "Value accessed through a second pointer: " << *p2 << '\n';
    std::cout << "Value accessed through a third pointer: " << *p3 << '\n';
}
```