1. Pass the following tests for a Builder design pattern. Note that pizza_director is a singleton.

   TEST1:
   ```
        pizza_director::register_pizza_builder( "Hawaii" ,
   &HawaiianPizzaBuilder::create);
        Pizza pizza = pizza_director::get_instance->getPizza("Hawaii");
        Dough my_dough = pizza.get_dough();
        Topping my_topping = pizza.get_topping();
        Sauce my_sauce = pizza.get_sauce();
        EXPECT_EQ(my_dough.brand(), "CrisspyFlower");
        EXPECT_EQ(my_topping.name(), "Pineapple");
        EXPECT_EQ(my_sauce.type(), "mayo");
   ```

   TEST2:
   ```
        pizza_director::register_pizza_builder( "SpicyCraz" ,
   &HawaiianPizzaBuilder::create );
        Pizza pizza = pizza_director::get_instance->getPizza("SpicyCraz");
        Dough my_dough = pizza.get_dough();
        Topping my_topping = pizza.get_topping();
        Sauce my_sauce = pizza.get_sauce();
        EXPECT_EQ(my_dough.brand(), "ToughFlower");
        EXPECT_EQ(my_topping.name(), "chili pepper");
        EXPECT_EQ(my_sauce.type(), "sour sauce");
   ```

2. Pass the following test for a Bridge design pattern

   ```
        Shape* triangle = new Triangle( new OpenGL_api );
        Shape* circle = new Circle( new DirectX_api );
        EXPECT_EQ(triangle->draw(), "using OpenGL to draw");
        EXPECT_EQ(circle->draw(), "using DirectX to draw");
        delete triangle;
        delete circle;
   ```

3. Use the same classes you wrote for problem set 2, make the following test work:

   ```
        NewShape* newtriangle = new AdapterTriangle( new Triangle ( new OpenGL_api) );
        EXPECT_EQ(newtriangle->newdraw(), "using OpenGL to draw");
        delete newtriangle;
   ```