1. A gas station needs to keep track of the number of gallons of gas it has on hand and the current price per gallon. The station has at least two pumps. Pumps 0 and 1 are full-service pumps, and all the rest are self-service. Self-service customers pay the base price for each gallon of gas, while full-service customers pay $0.25 more per gallon.

   Two classes are used to represent this situation. The `Pump` class handles the details for each pump and the `Station` class handles the overall gas station functions.

   Consider the following class declarations.

```
class Pump
{
  public:

    Pump();
    // postcondition: sets number of gallons sold at this pump to 0.0

    double GallonsSold() const;
    // postcondition: returns the number of gallons sold at this pump

    void ResetGallonsSold();
    // postcondition: resets number of gallons sold at this pump to 0.0

    // ... other public and private members not shown
};

class Station
{
  public:
    // constructor not shown

    double TotalSales() const;
    // postcondition: returns the total cash value of
    //                sales for all pumps

    void ResetAll();
    // postcondition: for every Pump p in this station
    //                p.GallonsSold() is 0.0

    void CloseStation(ostream & logFile);
    // precondition:  logFile is open and ready for writing
    // postcondition: writes the total cash value of
    //                all gas sold for the day to logFile;
    //                for every Pump p in this station
    //                p.GallonsSold() is 0.0

    // ... other public member functions not shown

  private:
    double myBasePrice;      // current price per gallon of gas
                             // for self-service pumps

    apvector<Pump> myPumps;  // the gas pumps; myPumps.length() > 1 and
                             // is the number of pumps in this station

    // ... other private data not shown
};
```

(a) Write the `Station` member function `ResetAll`, as started below. `ResetAll` changes the number of gallons sold at each pump to 0.0.

In writing `ResetAll`, you may call any of the public member functions of the `Pump` and `Station` classes. Assume that all these functions work as specified.

Complete function `ResetAll` below.

```
void Station::ResetAll()
// postcondition: For every Pump p in this station
//                p.GallonsSold() is 0.0
```

(b) Write the `Station` member function `TotalSales`, as started below. `TotalSales` returns the total cash value of the gallons sold at all pumps. Recall that self-service pumps charge the base price for each gallon of gas, while full-service pumps (pumps 0 and 1) charge $0.25 more per gallon.

In writing `TotalSales`, you may call any of the public member functions of the `Pump` and `Station` classes. Assume that all these functions work as specified.

Complete function `TotalSales` below.

```
double Station::TotalSales() const
// postcondition: returns the total cash value of sales
//                for all pumps
```

(c) Write the `Station` member function `CloseStation`, as started below. `CloseStation` will write the total amount of money earned from the day's gas sales to the output stream, `logFile`, and then reset the number of gallons sold at each individual pump to 0.0.

In writing `CloseStation`, you may call any of the public member functions of the `Pump` and `Station` classes. Assume that these functions, including `ResetAll` and `TotalSales`, work as specified, regardless of what you wrote in parts (a) and (b).

Complete function `CloseStation` below.

```
void Station::CloseStation(ostream & logFile)
// precondition:  logFile is open and ready for writing
// postcondition: writes the total cash value of
//                all gas sold for the day to logFile;
//                for every Pump p in this station
//                p.GallonsSold() is 0.0
```