1) Design a class named **MyInteger**. The class contains:

- An **int** data field named **value** that stores the **int** value represented by this object.
- A constructor that creates a **MyInteger** object for the specified **int** value.
- A **get** method that returns the **int** value.
- Methods **isEven()**, **isOdd()**, and **isPrime()** that return **true** if the value is even, odd, or prime, respectively.
- Static methods **isEven(int)**, **isOdd(int)**, and **isPrime(int)** that return **true** if the specified value is even, odd, or prime, respectively.
- Static methods **isEven(MyInteger)**, **isOdd(MyInteger)**, and **isPrime(MyInteger)** that return **true** if the specified value is even, odd, or prime, respectively.
- Methods **equals(int)** and **equals(MyInteger)** that return **true** if the value in the object is equal to the specified value.
- A static method **parseInt(char[])** that converts an array of numeric characters to an **int** value.
- A static method **parseInt(String)** that converts a string into an **int** value.

Draw the UML diagram for the class. Implement the class. Write a client program that tests all methods in the class.

2) Write a program that displays all the **prime numbers** less than **120** in decreasing order.

3) The class *Ticket* has been coded as follows:

```
public class Ticket

{

  private double price;

  private char service;


  public Ticket( double newPrice, char newService )

  {

    setPrice( newPrice );

    setService( newService );

  }

}
```

a) Write the *mutators (setPrice and setService methods)* for the Ticket class; the price must be greater than or equal to 0; the service must be either A or B, the default service is B.

b) Write the *accessor (getPrice and getService methods)* for the Ticket class.

c) Write a method that switches the value of service: if it is A, it changes to B; if it is B, it changes to A.

d) Change the *toString* method code to return the service and price separated by a : (colon) as in the following examples:

```
Example 1:      B:34.99

Example 2:      A:94.99

Example 3:      B:44.99
```

e) What is the data type of the parameter and the return type of the tax method?

```
public double tax( float rate )
```

f) In a client class and inside the *main* method, *myTicket* is an object reference of type *Ticket*; call the method *tax* with *myTicket,* assuming a tax rate of 0.06, and assign the resulting tax value to a variable named *myTax*.

```
float taxRate = 0.06f;
```

g) Consider the following constant of class *Ticket*
```
public static char DEFAULT_SERVICE = 'B';
```

In a client class and inside *main*, write a statement to output the value of the above constant.