Extra Credit Assignment 2

parseBinary() function	-- 25 pts.
TwoDPoint and ThreeDPoint Classes	-- 25 pts.

parseBinary() function

- Write a function that parses a binary number as a string into a decimal integer. The function header is as follows:
  `int parseBinary(char* binaryString)`
  For example, binary string 10001 is 17:
  $1*2^4+ 0*2^3+1*2^2+0*2^1+1*2^0 = 16+1 = 17$
  So, `parseBinary("10001")` returns 17.
- Make `parseBinary()` function throw an exception of type integer (like -1) in case if the string that is being parsed is not a binary number (contains characters other that 0 or1).
- In main() ask the user to enter a binary number, pass the string given by the user into the function. Surround `parseBinary()` function calls with try block. Write catch block that catches exceptions of type integer. Test-run the code on binary strings of different length. Make sure the exception is being thrown and caught.

2DPoint and 3DPoint Classes

1. Design a class named TwoDPoint to represent a point with x and y coordinates. The class contains:
   - Two data fields x and y that represent the coordinates.
   - Default constructor that creates a point (0,0)
   - A constructor that constructs a point with specified coordinates
   - Two accessor functions
   - A function named distance that returns the distance from this point to the origin. Function header:

   ```
   virtual double distance() const
   ```

   Formula for calculating distance between two-dimensional point (x1, y1) and the origin is:

   ```
   d = sqrt( (x1)^2+(y1)^2)
   ```

2. Create a class named ThreeDPoint to model a point in a three-dimensional space. let ThreeDPoint be derived from TwoDPoint with the following additional features:
   - A data field z that represents the z-coordinate.
   - Default constructor that creates a point (0,0,0)
   - Accessor function for z

- Override the `distance` function to return the distance between the point in the three-dimensional space and the origin. Function header:

```
double distance() const
```

Formula for calculating this distance is

$$d = \text{sqrt}((x1)^2+(y1)^2+(z1)^2)$$

where the coordinates of the point are (x1, y1, z1)

3. Test your new classes in main(). Specifically, make sure that the following code calls the correct implementation of the distance function:

```
TwoDPoint * ptPtr1 = new TwoDPoint(3, 5);
TwoDPoint * ptPtr2 = new ThreeDPoint(7, 10, 15);
cout<<"The 2D distance is "<< ptPtr1->distance();
cout<<"The 3D distance is "<< ptPtr1->distance();
```