```
for(int i = 0; i<messageChar.length;i++)
{
    revMessage.add(messageChar[messageChar.length-i]) ;
}
```

Logic error: the array is trying to input a value into an undefined index

```
for(int i = messageChar.length-1; i>=0;i--)
{
    revMessage.add(messageChar[i]) ;
}
```

Fixed by decreasing i and thus keeping the index of messageChar to be added as i.

```
        for(int j =0; j< messageChar.length;j++)
        {
            if(revMessage.get(j) != messageChar[j])
            {
                System.out.println("Your phrase is NOT a Palindrome.");
                break;
            }
            else
            {
                System.out.println("Your phrase IS a Palindrome.");
                break;
            }
        }
    }
}
```

Logic error: the “is a palindrome” statement only accounts for if the first and last letter of the phrase are the same rather than the entire phrase.

```
        for(int j =0; j< messageChar.length;j++)
        {
            if(revMessage.get(j) != messageChar[j])
            {
                pdrome = false;
                break;
            }
            else
            {
                pdrome = true;
            }
        }

        if (pdrome = false)
        {
            System.out.print("Your message is NOT a Palindrome.");
        }

        else
        {
            System.out.print("The message IS a Palindrome.");
        }
```

Syntax error: in if statement i am using the sign to make pdrome equal to false rather than comparing it.

```
        boolean pdrome = false;

        for(int j =0; j< messageChar.length;j++)
        {
            if(revMessage.get(j) != messageChar[j])
            {
                pdrome = false;
                break;
            }
            else
            {
                pdrome = true;
            }
        }

        if (pdrome == false)
        {
            System.out.print("Your message is NOT a Palindrome.");
        }

        else
        {
            System.out.print("The message IS a Palindrome.");
        }
```

Both issues are fixed here by using a boolean variable to hold true as long as reverse character is the same as original message character.