Java Assignment - 2

Validate Enrollment Number:

```
class ValidateEnroll
{
	static boolean validateString(String enrollString)
	{
		boolean valid = false;
		String yearString = enrollString.substring(0,2);
		String collegeString = enrollString.substring(2,5);
		String numString = enrollString.substring(5,7);
		String branchString = enrollString.substring(7,9);
		String rollString = enrollString.substring(9,12);
		int rollNumber = Integer.parseInt(rollString);
		if(yearString.equals("16") | yearString.equals("17") |
                                    yearString.equals("18"))
			if(collegeString.equals("047"))
				if(numString.equals("01") |
                                        numString.equals("31"))
                                    if(branchString.equals("07"))
                                        if(rollNumber >=0 & rollNumber<=70)
                                            valid = true;
		return valid;
	}

	public static void main(String args[])
	{
		//args[0] is 170470107023
		boolean valid = false;
		valid = validateString(args[0]);
		if(valid)
			System.out.println("Enrollment Number is valid.");
		else
			System.out.println("Invalid Enrollment Number.");
	}
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEnroll>javac ValidateEnroll.java

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEnroll>java ValidateEnroll 160470107023
Enrollment Number is valid.

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEnroll>java ValidateEnroll 190470107088
Invalid Enrollment Number.
```

Validate e-mail Address:

```
public class ValidateEmail {

	static boolean checkFirstCharacter(String emailAddress)
	{
		char firstCharacter = emailAddress.charAt(0);
		if((firstCharacter >= 'a' & firstCharacter <= 'z')|
				(firstCharacter >= '0' & firstCharacter<=’9’))
			return true;
		else
			return false;
	}

	static boolean notAtSymbol(String emailAddress)
	{
		boolean containsAtIndex = emailAddress.contains("@");
		if(!containsAtIndex)
			return true;
		else
			return false;
	}

	static boolean notDotSymbol(String emailAddress)
	{
		int atIndex;
		boolean containsDot = false;
		boolean containsAtIndex = emailAddress.contains("@");
		if(containsAtIndex)
		{
			atIndex = emailAddress.indexOf("@");
			String dotSubstring =
					emailAddress.substring(atIndex);
			containsDot = dotSubstring.contains(".");
		}
		if(!containsDot)
			return true;
		else
			return false;
	}
```

```
static boolean checkUsernameLength(String emailAddress)
{
      boolean containsAtIndex = emailAddress.contains("@");
      if(containsAtIndex)
      {
            String[] atSplit = emailAddress.split("@");
            //atSplit[0] should be username: mihir67mj
            int usernameLength = atSplit[0].length();
            if(usernameLength >= 8 & usernameLength <= 20)
                  return true;
            else
                  return false;
      }
      else return false;
}

static boolean mailServiceLength(String emailAddress)
{
      boolean containsAtIndex = emailAddress.contains("@");
      boolean containsDot = emailAddress.contains(".");
      if(containsAtIndex)
      {
            if(containsDot)
            {
                  int atIndex = emailAddress.indexOf("@");
                  int dotIndex = emailAddress.indexOf(".",
                                                      atIndex);
                  String domainName =
                   emailAddress.substring(atIndex + 1,dotIndex);
                  int domainLength = domainName.length();
                  if(domainLength >= 2)
                        return true;
                  else
                        return false;
            }
            else return false;
      }
      else return false;
}
```

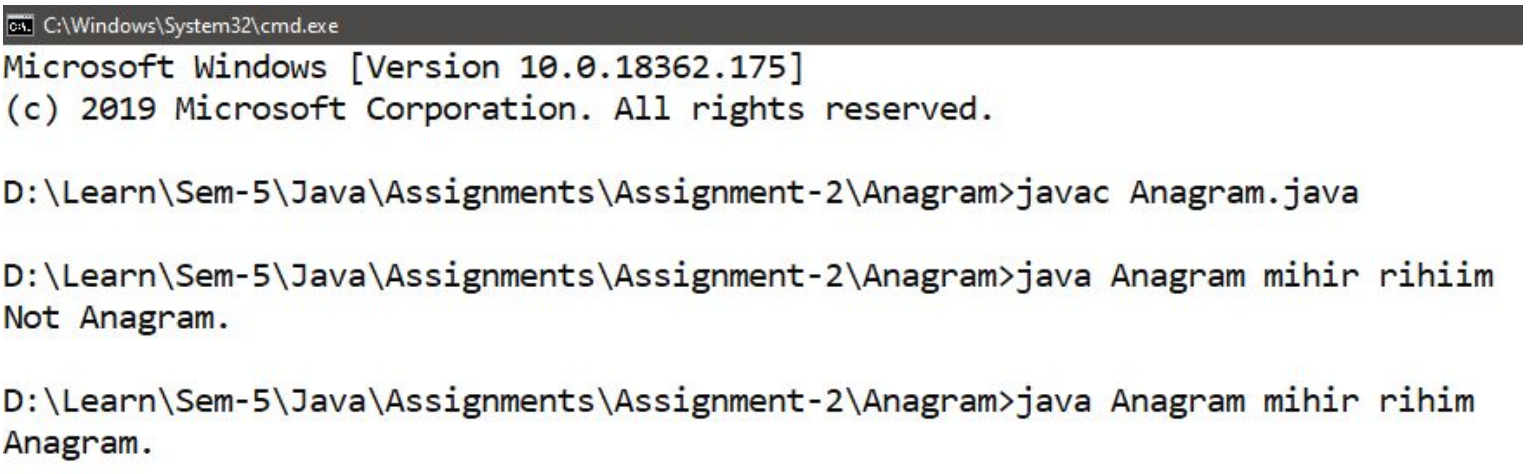
```
static boolean invalidSpecialChar(String emailAddress)
{
      boolean containsAtIndex = emailAddress.contains("@");
      if(containsAtIndex)
      {
            String[] atSplit = emailAddress.split("@");
            boolean invalid = false;
            //at this point we only need to check if username
                                          contains '_' or '.'
            //as the only available special characters
            //atSplit[0] = "mihir67mj"
            char[] username = atSplit[0].toCharArray();
            for(int i=0; i<atSplit[0].length(); i++)
            {
                  if( !((username[i] >= 'a' & username[i] <='z')
                  |(username[i] == '_' | username[i] == '.')
                  |(username[i] >= '0' & username[i] <= '9')) )
                  {
                        System.out.println(username[i]);
                        invalid = true;
                        break;
                  }
            }
            return invalid;
      }
      else return true;

}

static boolean invalidDomain(String emailAddress)
{
      boolean containsAtIndex = emailAddress.contains("@");
      boolean containsDot = emailAddress.contains(".");
      if(containsAtIndex)
            if(containsDot)
            {
                  int dotIndex = emailAddress.indexOf(".");
                  String domainName =
                              emailAddress.substring(dotIndex);
                  int domainLength = domainName.length();
                  if(domainLength >= 3)
                        return false;
                  else
                        return true;
            }
      return true;   }
```

```
public static void main(String args[])
{
      //args[0] contains an e-mail address
      //mihir67mj@gmail.com
      String emailAddress = args[0];
      String errorMessage = "";
      if(!checkFirstCharacter(emailAddress))
      {
            errorMessage += ("Enter a valid first
                                          character.\n");
            char firstCharacter = emailAddress.charAt(0);
            if(firstCharacter >= 'A' & firstCharacter <= 'Z')
                  errorMessage += ("The first character can't be
                                          in Uppercase.\n");
            else
                  errorMessage += ("The first character can't be
                                       a special symbol.\n");
      }
      if(notAtSymbol(emailAddress))
            errorMessage += ("Enter an e-mail with @\n");
      if(notDotSymbol(emailAddress))
            errorMessage += ("Enter an e-mail with . after
                                                   @\n");
      if(!checkUsernameLength(emailAddress))
      {
            errorMessage += ("Enter a valid username.\n");
            errorMessage += ("Username length must be between
                                             8-20.\n");
      }
      if(!mailServiceLength(emailAddress))
            errorMessage += ("MailServiceName must be at least 2
                                       characters long.\n");
      if(invalidSpecialChar(emailAddress))
            errorMessage += ("Enter a valid username: only a-z,
                        underscore and dot are allowed.\n");
      if(invalidDomain(emailAddress))
      {
            errorMessage += ("Enter a valid domain.\n");
            errorMessage += ("e.g.: .com or .org\n");
      }
      if(errorMessage.length()==0)
            System.out.println("Congratulations. Your E-mail is
                                             valid.");
      else
            System.out.println(errorMessage);
} }
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEmail>javac ValidateEmail.java

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEmail>java ValidateEmail mihir67mj
Enter an e-mail with @
Enter an e-mail with . after @
Enter a valid username.
Username length must be between 8-20.
MailServiceName must be at least 2 characters long.
Enter a valid username: only a-z, underscore and dot are allowed.
Enter a valid domain.
e.g.: .com or .org


D:\Learn\Sem-5\Java\Assignments\Assignment-2\ValidateEmail>java ValidateEmail mihir67mj@gmail.com
Congratulations. Your E-mail is valid.
```

Anagram String:

```
class Anagram {
	public static void main(String args[])
	{
		boolean notAnagram = false;
		String testString = args[0];
		String testString2 = args[1];
		int[] arrA = new int[26];
		int[] arrB = new int[26];
		for(int i = 0; i < testString.length(); i++)
		{
			char c = testString.charAt(i);
			arrA[c - 96]++;
		}
		for(int i = 0; i < testString2.length(); i++)
		{
			char c = testString2.charAt(i);
			arrB[c - 96]++;
		}
		for(int i = 0;i<26;i++)
		{
			if(arrA[i] != arrB[i])
			{
				notAnagram = true;
				break;
			}
		}
		if(notAnagram)
			System.out.println("Not Anagram.");
		else
			System.out.println("Anagram.");
	}

}
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.
D:\Learn\Sem-5\Java\Assignments\Assignment-2\Anagram>javac Anagram.java
D:\Learn\Sem-5\Java\Assignments\Assignment-2\Anagram>java Anagram mihir rihiim
Not Anagram.
D:\Learn\Sem-5\Java\Assignments\Assignment-2\Anagram>java Anagram mihir rihim
Anagram.

Reverse Words:

```
class ReverseWords {
      public static void main(String args[])
      {
            String cName = args[0];
            String[] cSplit = cName.split(" ");
            StringBuffer[] finalStr = new
StringBuffer[cSplit.length];
            StringBuffer answer = new StringBuffer();
            for(int i = 0; i < cSplit.length ; i++)
            {
                  finalStr[i] = new StringBuffer(cSplit[i]);
                  finalStr[i].reverse();
                  answer.append(finalStr[i]);
                  answer.append(" ");
            }
            System.out.println(answer);
      }

}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ReverseWords>javac ReverseWords.java

D:\Learn\Sem-5\Java\Assignments\Assignment-2\ReverseWords>java ReverseWords "vvp engineering college"
pvv gnireenigne egelloc
```