ASSIGNMENT

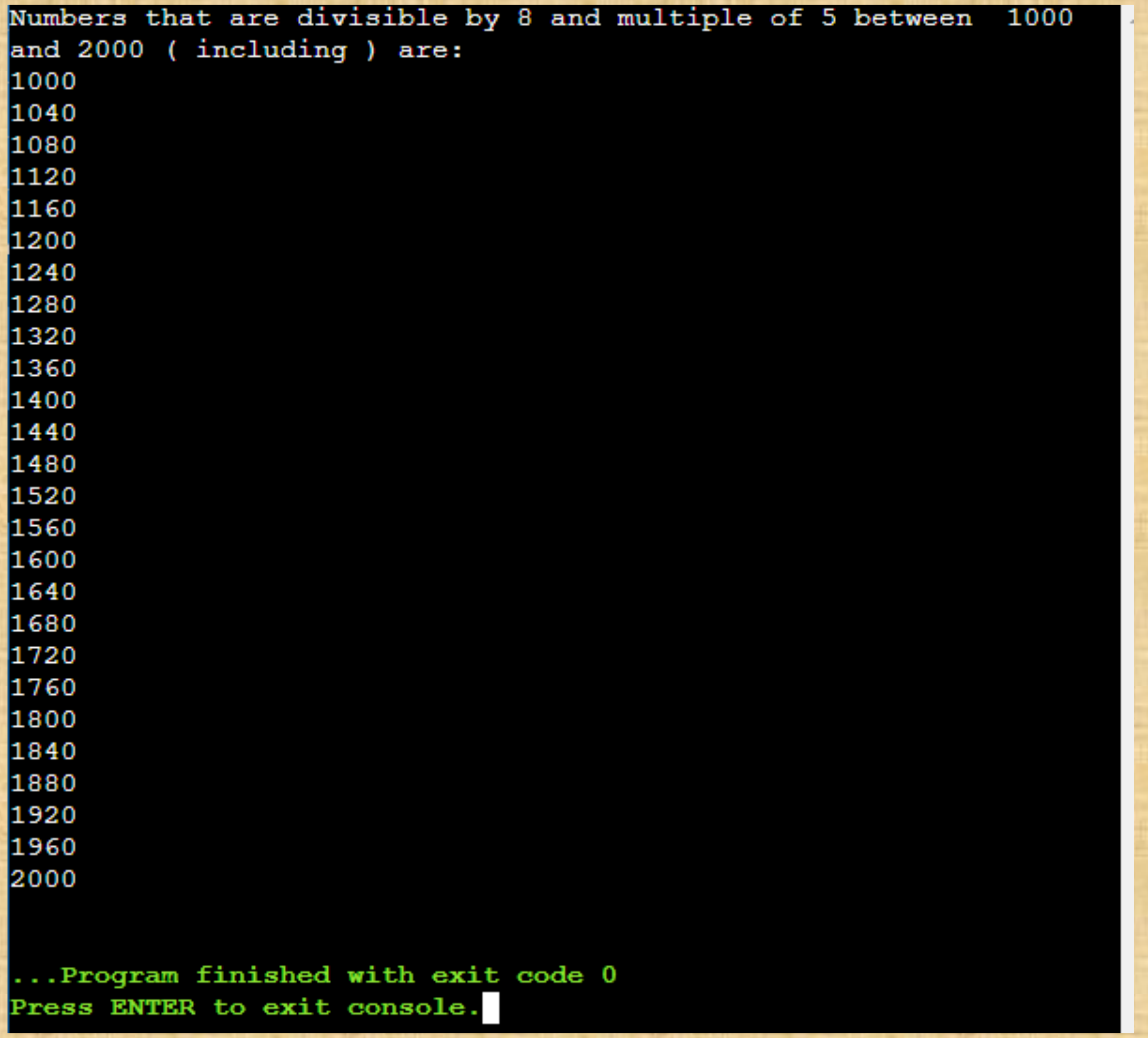
TUTORIAL 02

Q1) Write a JAVA program to find those numbers which are divisible by 8 and multiple of 5, between 1000 and 2000 (both included)

CODE:

```
public class DivisibleAndMultiple{
   public static void main(String[] args){
      System.out.println("Numbers that are divisible by 8 and multiple of 5
between  1000  and 2000 ( including ) are:");
      for(int n=1000;n<=2000;n++){
         if(n%8==0 && n%5==0){
            System.out.println(n+ " ");
         }
      }
   }
}
```

Output:



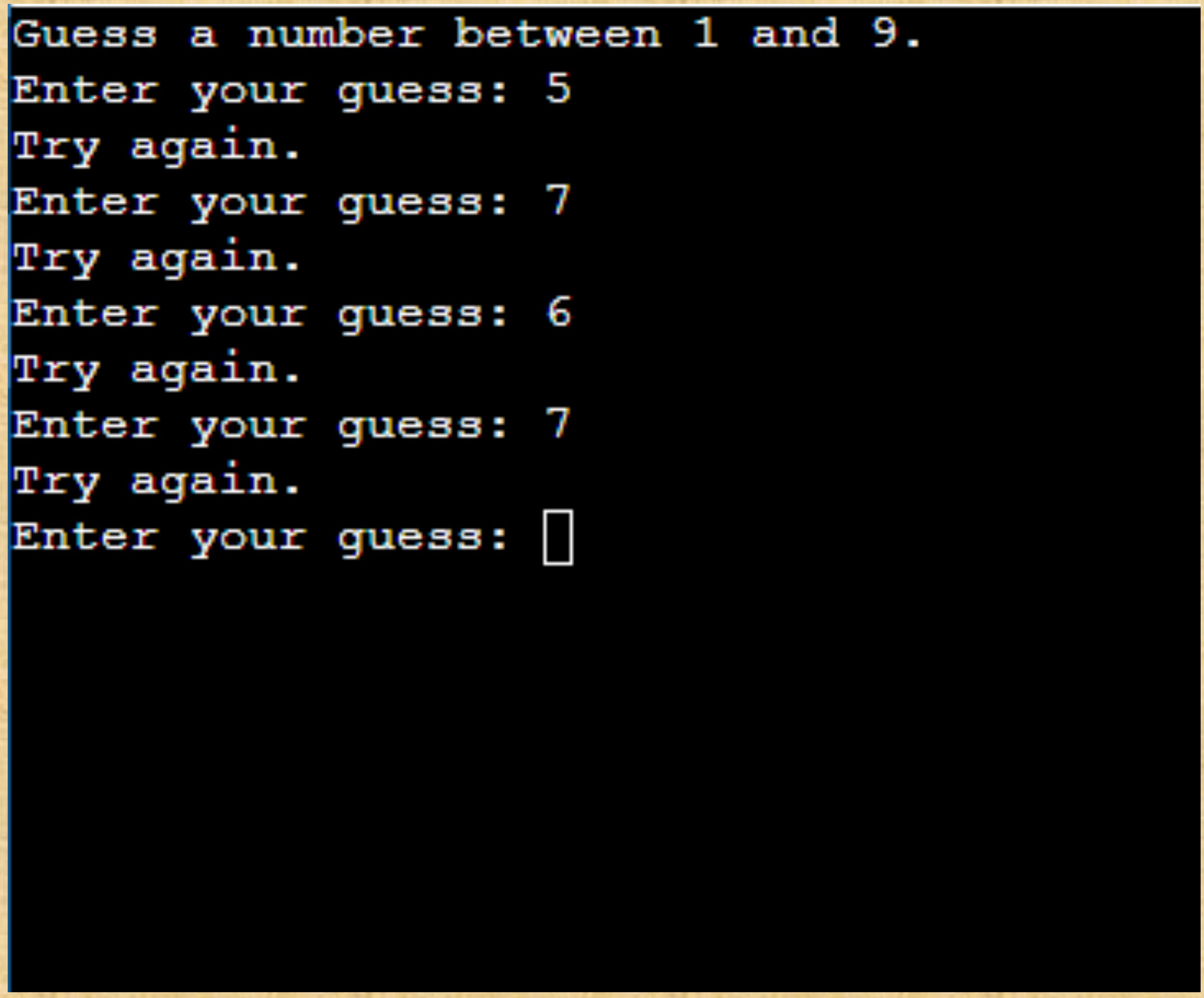
```
Numbers that are divisible by 8 and multiple of 5 between  1000
and 2000 ( including ) are:
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000


...Program finished with exit code 0
Press ENTER to exit console.
```

Q2) Write a JAVA program to guess a number between 1 to 9. Note: User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a "Well guessed!" message, and the program will exit.

CODE:

```
import java.util.Scanner;
import java.util.Random;

public class GuessNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();

        int targetNumber = random.nextInt(9) + 1;
        int userGuess;

        System.out.println("Guess a number between 1 and 9.");

        do {
            System.out.print("Enter your guess: ");
            userGuess = scanner.nextInt();

            if (userGuess == targetNumber) {
                System.out.println("Well guessed!");
                break;
            } else {
                System.out.println("Try again.");
            }
        } while (true);

        scanner.close();
    }
}
```

OUTPUT



```
Guess a number between 1 and 9.
Enter your guess: 5
Try again.
Enter your guess: 7
Try again.
Enter your guess: 6
Try again.
Enter your guess: 7
Try again.
Enter your guess:
```

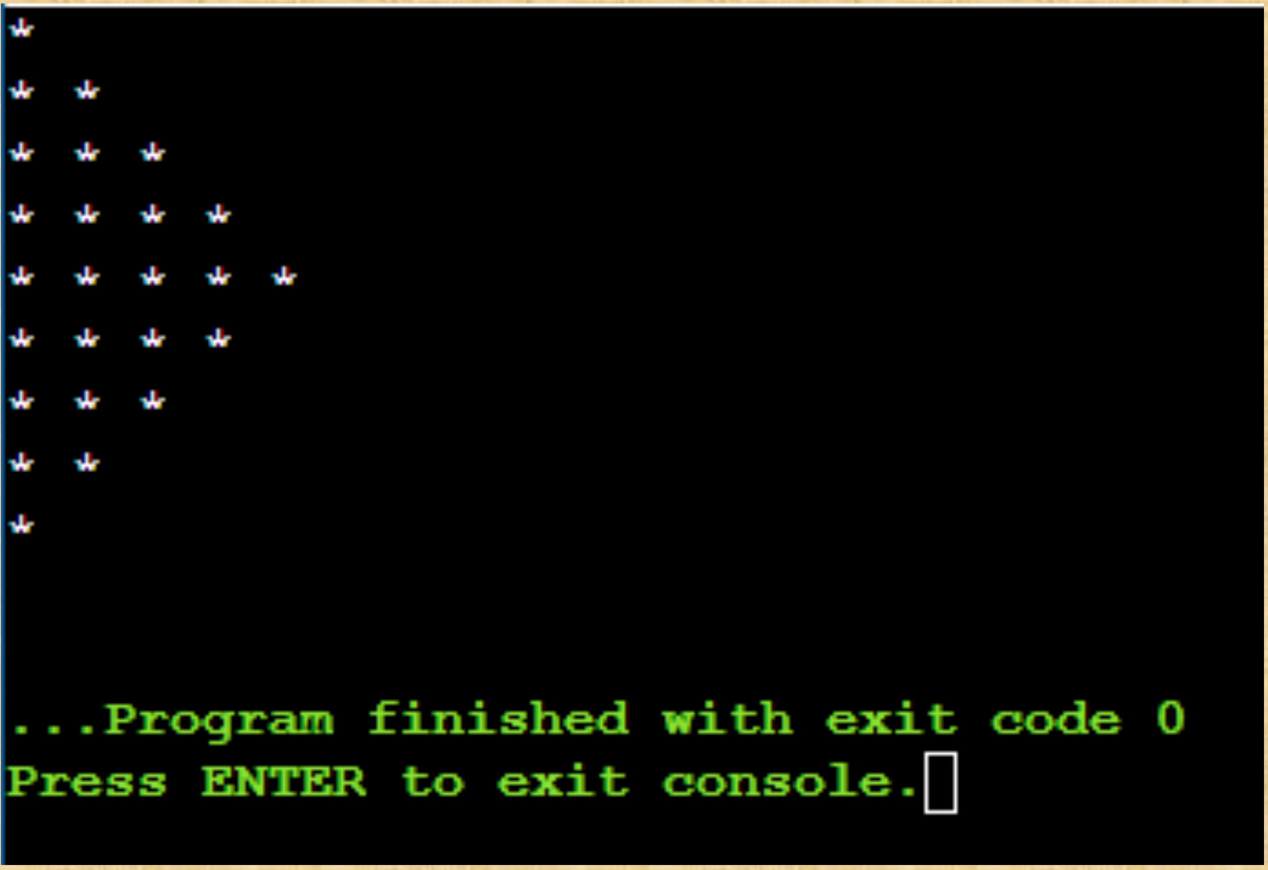
Q3) Write a JAVA program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

CODE:

```
public class Pattern {
    public static void main(String[] args) {
        int rows = 5;  // Number of rows in the pattern

        // Print upper half of the pattern
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }

        // Print lower half of the pattern
        for (int i = rows - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:



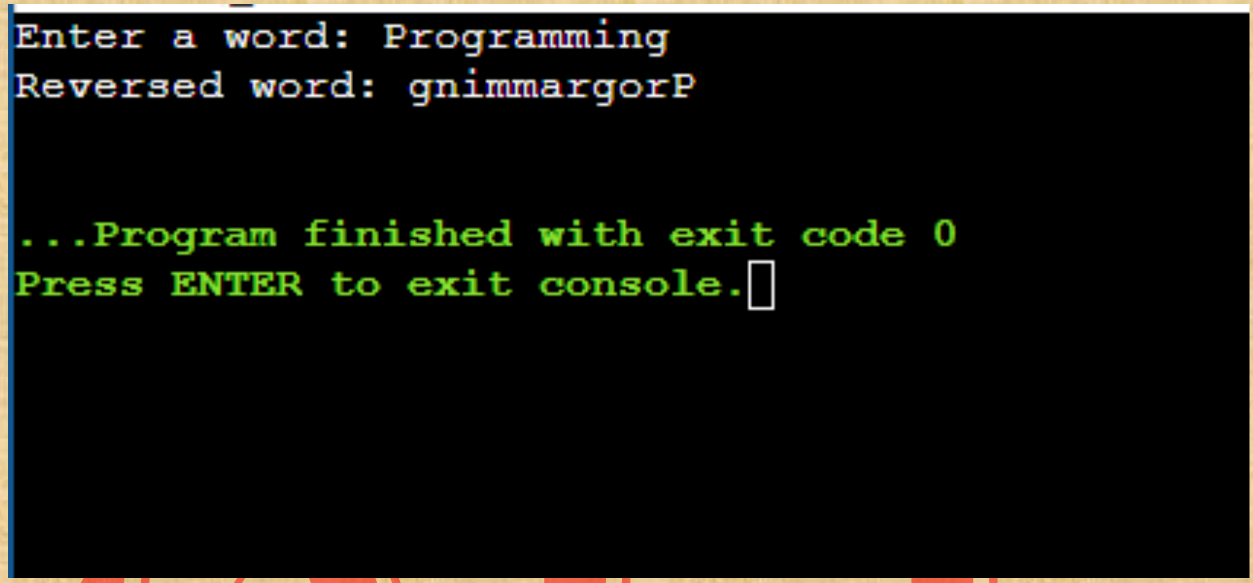
```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *
*  *  *
*  *
*


...Program finished with exit code 0
Press ENTER to exit console.
```

Q4) Write a JAVA program that accepts a word from the user and reverse it. (should not use any functions)

CODE:

```java
import java.util.Scanner;

public class ReverseWord {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = scanner.nextLine();

        String reversedWord = reverseWord(word);
        System.out.println("Reversed word: " + reversedWord);

        scanner.close();
    }

    public static String reverseWord(String word) {
        char[] charArray = word.toCharArray();
        int left = 0;
        int right = charArray.length - 1;

        while (left < right) {
            char temp = charArray[left];
            charArray[left] = charArray[right];
            charArray[right] = temp;

            left++;
            right--;
    }

    return new String(charArray);
  }
}
```

OUTPUT:



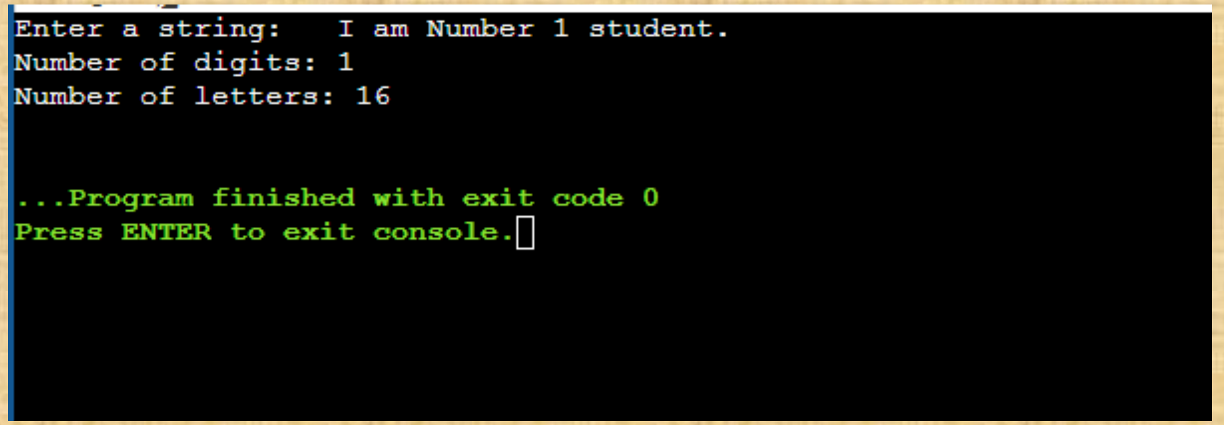
```
Enter a word: Programming
Reversed word: gnimmargorP


...Program finished with exit code 0
Press ENTER to exit console.
```

Q5) Write a JAVA program that accepts a string and calculate the number of digits and letters.

CODE

```
import java.util.Scanner;

public class CountDigitsAndLetters {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a string: ");
    String input = scanner.nextLine();

    int digitCount = 0;
    int letterCount = 0;

    for (int i = 0; i < input.length(); i++) {
      char ch = input.charAt(i);
      if (Character.isDigit(ch)) {
        digitCount++;
      } else if (Character.isLetter(ch)) {
        letterCount++;
      }
    }

    System.out.println("Number of digits: " + digitCount);
    System.out.println("Number of letters: " + letterCount);

    scanner.close();
  }
}
```

OUTPUT:



```
Enter a string:   I am Number 1 student.
Number of digits: 1
Number of letters: 16


...Program finished with exit code 0
Press ENTER to exit console.
```

Q6) Write a JAVA program to check the validity of password input by users.

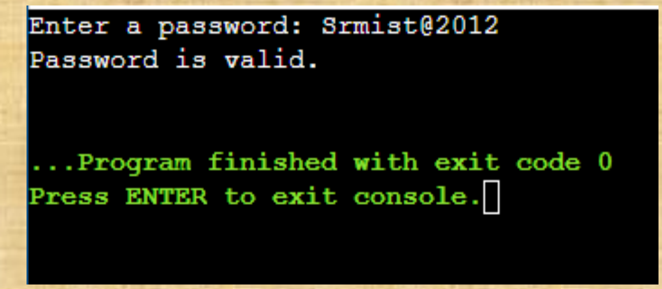
Validation:

- **At least 1 letter between [a-z] and 1 letter between [A-Z].**
- **At least 1 number between [0-9].**
- **At least 1 character from [$#@].**
- **Minimum length 6 characters.**
- **Maximum length 16 characters.**

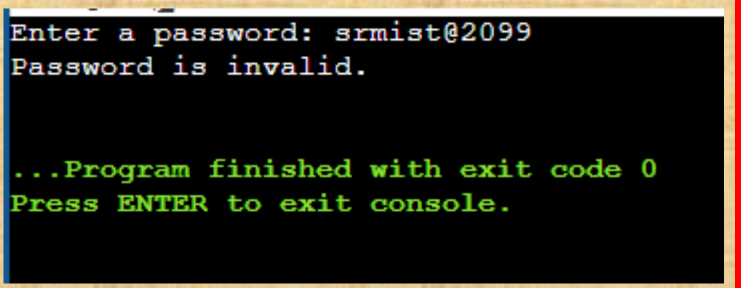
CODE:

```
import java.util.Scanner;

public class PasswordValidation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a password: ");
        String password = scanner.nextLine();
        if (isValidPassword(password)) {
            System.out.println("Password is valid.");
        } else {
            System.out.println("Password is invalid.");
        }

        scanner.close();
    }

    public static boolean isValidPassword(String password) {
        if (password.length() < 6 || password.length() > 16) {
            return false;
        }

        boolean hasLower = false;
        boolean hasUpper = false;
        boolean hasDigit = false;
        boolean hasSpecial = false;
        String specialChars = "$#@";

        for (char ch : password.toCharArray()) {
            if (Character.isLowerCase(ch)) {
                hasLower = true;
            } else if (Character.isUpperCase(ch)) {
                hasUpper = true;
            } else if (Character.isDigit(ch)) {
                hasDigit = true;
            } else if (specialChars.indexOf(ch) != -1) {
                hasSpecial = true;
            }
        }

        return hasLower && hasUpper && hasDigit && hasSpecial;
    }
}
```

OUTPUT:



```
Enter a password: Srmist@2012
Password is valid.


...Program finished with exit code 0
Press ENTER to exit console.
```



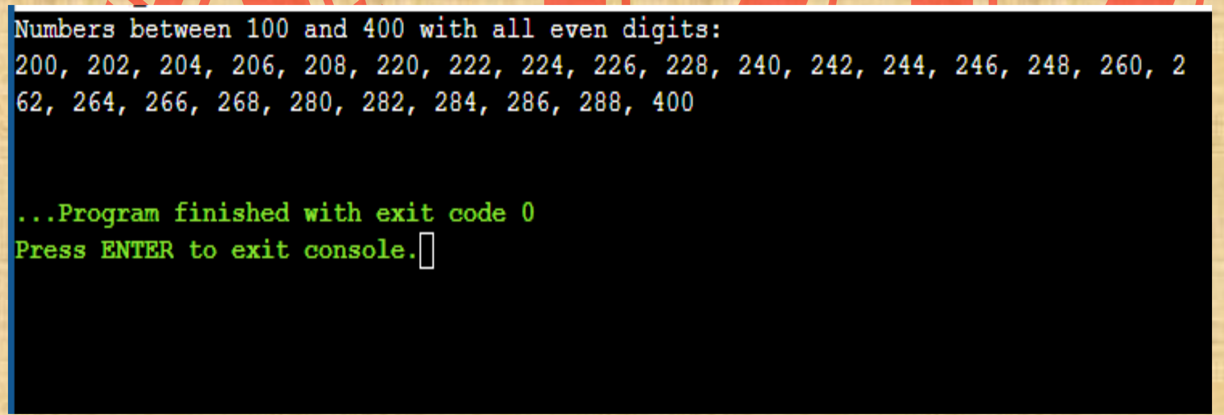
```
Enter a password: srmist@2099
Password is invalid.


...Program finished with exit code 0
Press ENTER to exit console.
```

Q7) Write a JAVA program to find numbers between 100 and 400 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

CODE:

```
public class EvenDigitNumbers {
    public static void main(String[] args) {
        System.out.println("Numbers between 100 and 400 with all even digits:");
        boolean first = true;

        for (int number = 100; number <= 400; number++) {
            if (hasOnlyEvenDigits(number)) {
                if (!first) {
                    System.out.print(", ");
                }
                System.out.print(number);
                first = false;
            }
        }

        System.out.println();
    }

    public static boolean hasOnlyEvenDigits(int number) {
        while (number > 0) {
            int digit = number % 10;
            if (digit % 2 != 0) {
                return false;
            }
            number /= 10;
        }
        return true;
    }
}
```

<u>OUTPUT:</u>



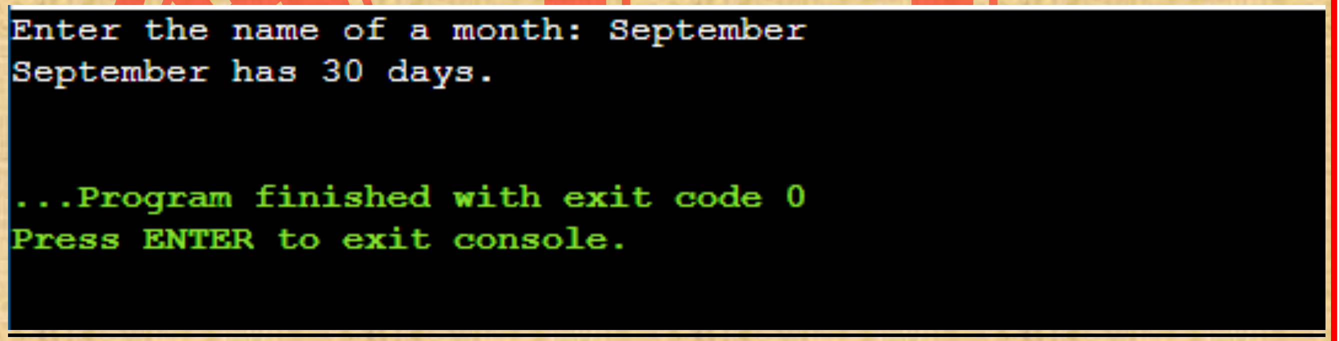
```
Numbers between 100 and 400 with all even digits:
200, 202, 204, 206, 208, 220, 222, 224, 226, 228, 240, 242, 244, 246, 248, 260, 2
62, 264, 266, 268, 280, 282, 284, 286, 288, 400


...Program finished with exit code 0
Press ENTER to exit console.
```

Q8) Write a JAVA program to convert month name to a number of days.

CODE:

```
import java.util.Scanner;

public class MonthToDaysConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the name of a month: ");
        String monthName = scanner.nextLine();

        int days = getDaysInMonth(monthName);

        if (days != -1) {
            System.out.println(monthName + " has " + days + " days.");
        } else {
            System.out.println("Invalid month name.");
        }

        scanner.close();
    }

    public static int getDaysInMonth(String monthName) {
        String[] months = {
            "January", "February", "March", "April",
            "May", "June", "July", "August",
            "September", "October", "November", "December"
        };

        int[] daysInMonths = {
            31, 28, 31, 30,
            31, 30, 31, 31,
            30, 31, 30, 31
        };

        for (int i = 0; i < months.length; i++) {
            if (months[i].equalsIgnoreCase(monthName)) {
                return daysInMonths[i];
            }
        }

        return -1; // Invalid month name
    }
}
```

OUTPUT:



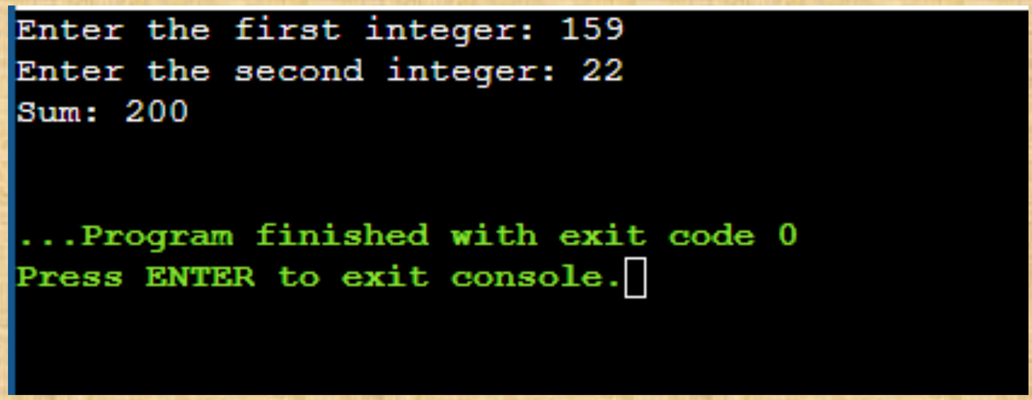
```
Enter the name of a month: September
September has 30 days.


...Program finished with exit code 0
Press ENTER to exit console.
```

Q9) Write a JAVA program to sum of two given integers. However, if the sum is between 105 to 200 it will return 200.

CODE:

```
import java.util.Scanner;

public class SumWithRangeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first integer: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second integer: ");
        int num2 = scanner.nextInt();
        int sum = num1 + num2;
        if (sum >= 105 && sum <= 200) {
            sum = 200;
        }
        System.out.println("Sum: " + sum);
        scanner.close();
    }
}
```

OUTPUT:



```
Enter the first integer: 159
Enter the second integer: 22
Sum: 200


...Program finished with exit code 0
Press ENTER to exit console.
```

Q 10) Write a JAVA program to construct the following pattern, using a nested loop number.

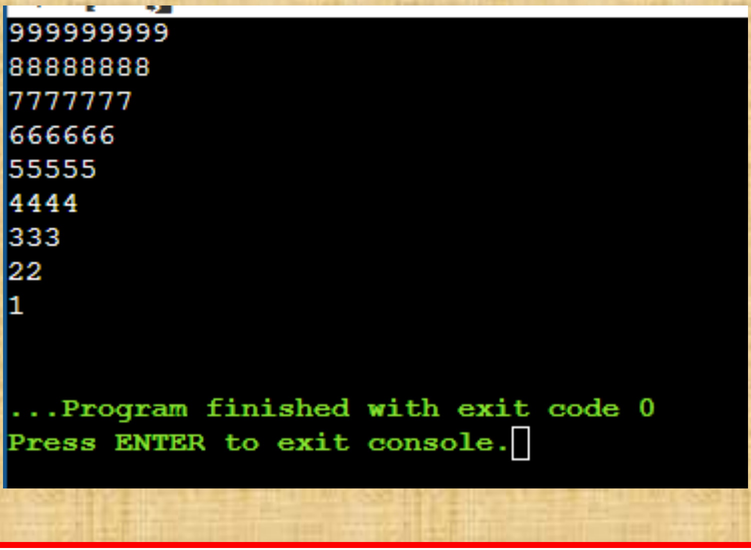
Expected Output:

```
999999999
88888888
7777777
666666
55555
4444
333
22
1
```

CODE:

```
public class NestedLoopPattern {
    public static void main(String[] args) {
        int rows = 9;  // Number of rows in the pattern
        for (int i = rows; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

OUTPUT:



```
999999999
88888888
7777777
666666
55555
4444
333
22
1


...Program finished with exit code 0
Press ENTER to exit console.
```