Name – Adya Singh

Reg No – RA2211003010181

21CSC203P

Advanced Programming Practice

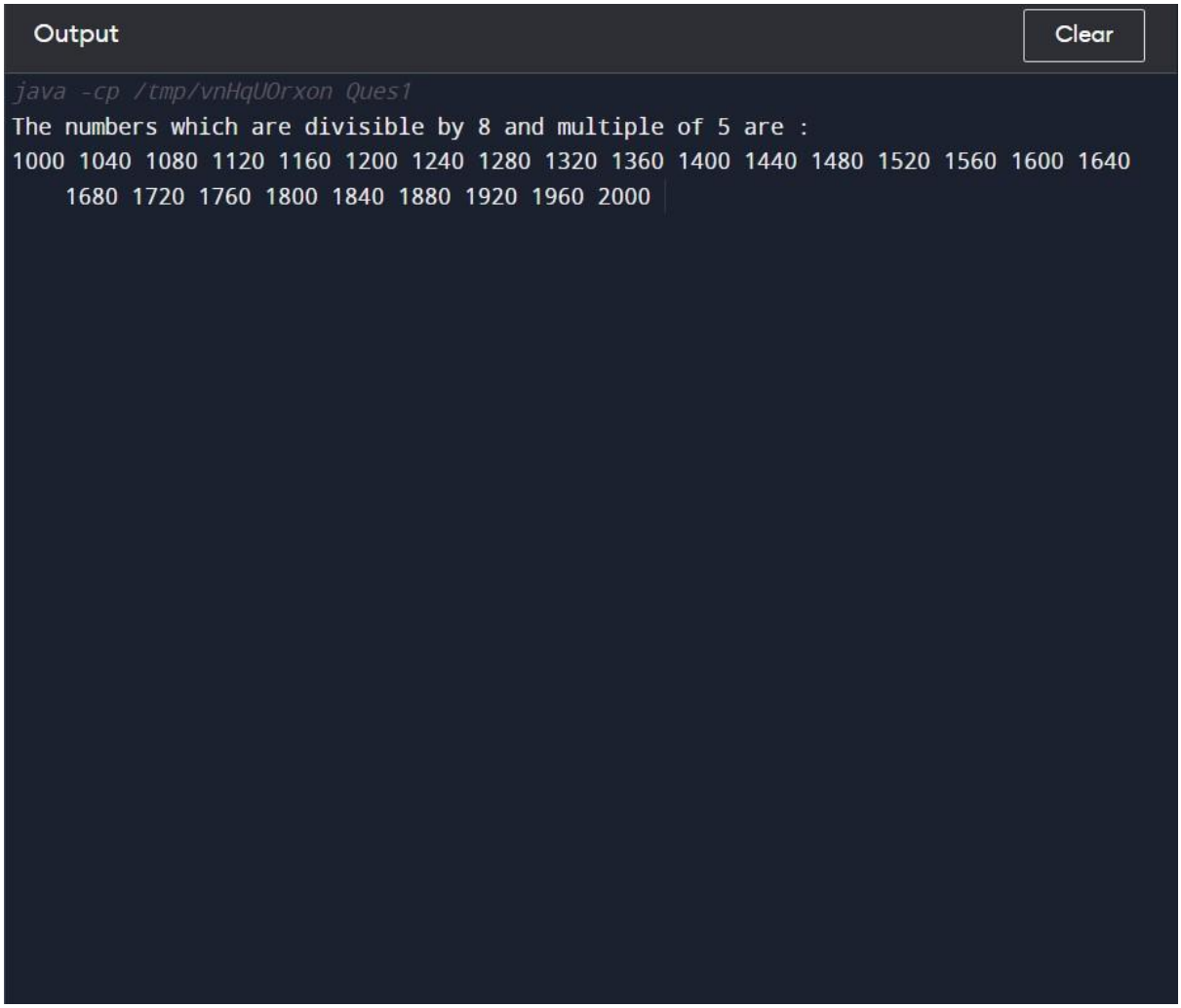
Assignment 4

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

Code –

```
class Number {

public static void main(String[] args) {

int i;

System.out.println("The numbers which are divisible by 8 and multiple of 5 are : ");

        for(i=1000; i<=2000;i++)

       {

          if(i%8==0 && i%5==0)

          System.out.print(i+" ");

       }

    }

}
```

Output –

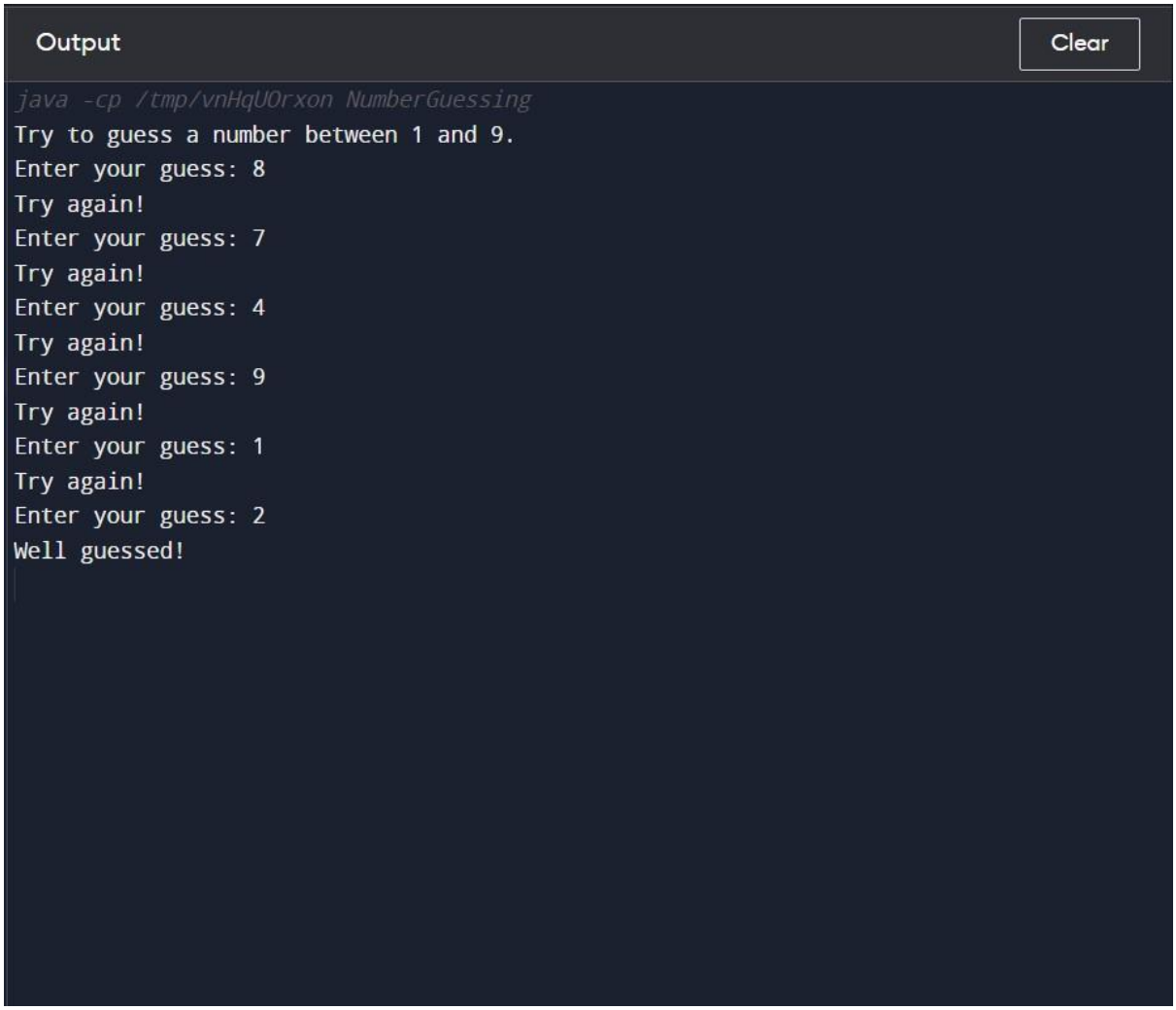

Output
Clear
java -cp /tmp/vnHqUOrxon Ques1
The numbers which are divisible by 8 and multiple of 5 are :
1000 1040 1080 1120 1160 1200 1240 1280 1320 1360 1400 1440 1480 1520 1560 1600 1640
1680 1720 1760 1800 1840 1880 1920 1960 2000

2. 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;

public class NumberGuessing {

   public static void main(String[] args) {

      int targetNumber = (int) (Math.random() * 9) + 1; // Generates a random number between 1 and 9

      Scanner scanner = new Scanner(System.in);

      int guess;

      System.out.println("Try to guess a number between 1 and 9.");

      do

   {

         System.out.print("Enter your guess: ");

         guess = scanner.nextInt();

         if (guess == targetNumber) {

            System.out.println("Well guessed!"); }

         else {

            System.out.println("Try again!"); }

      }

while (guess != targetNumber);

      scanner.close();

   }

}
```

Output –


Output
Clear
java -cp /tmp/vnHqUOrxon NumberGuessing
Try to guess a number between 1 and 9.
Enter your guess: 8
Try again!
Enter your guess: 7
Try again!
Enter your guess: 4
Try again!
Enter your guess: 9
Try again!
Enter your guess: 1
Try again!
Enter your guess: 2
Well guessed!

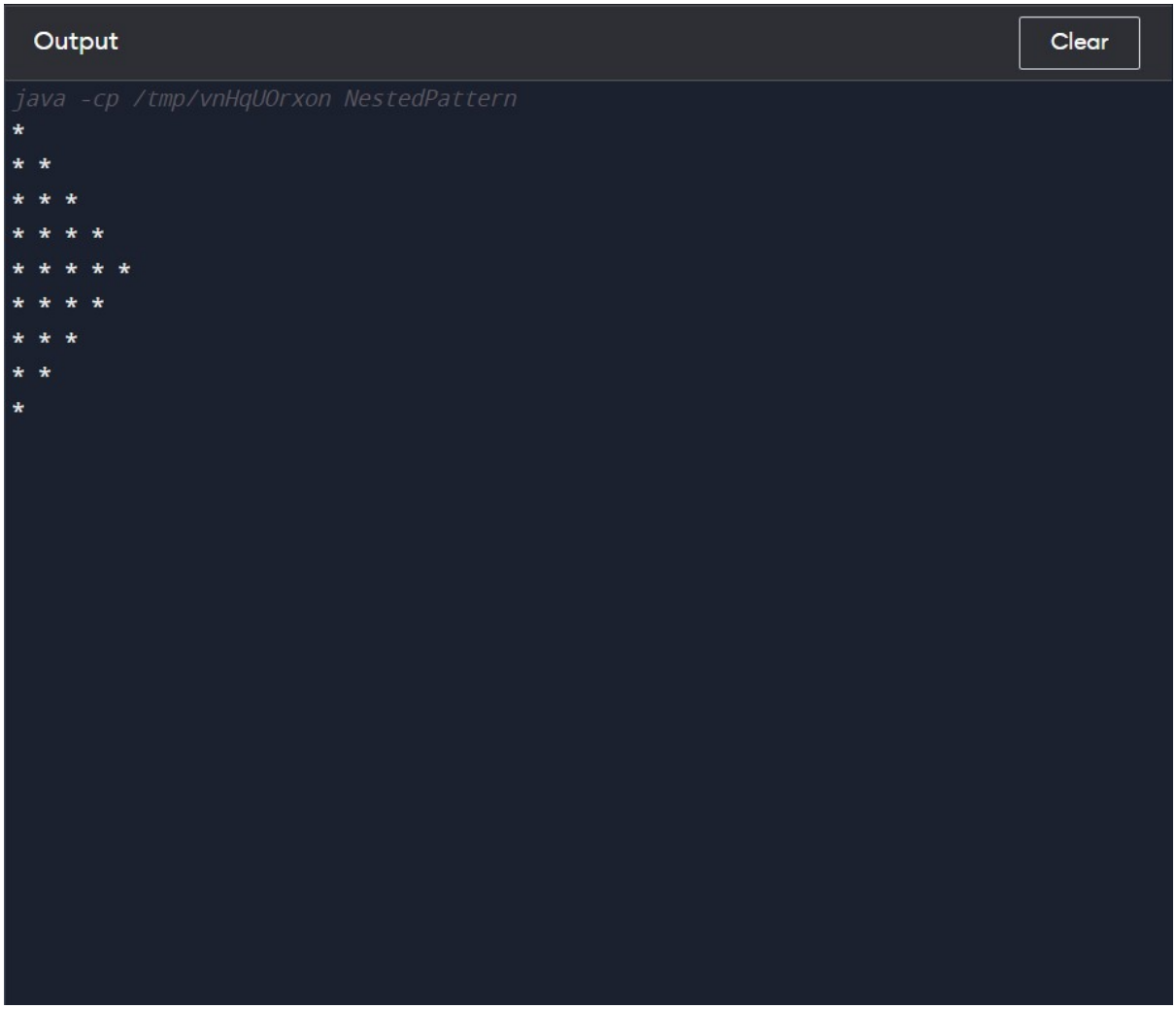
3. Write a JAVA program to construct the following pattern, using a nested for loop.

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

Code –

```
public class Pattern {
    public static void main(String[] args) {
        int rows = 5;
      for (int i = 1; i <= rows; i++) {          // Upper half of the pattern
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
            for (int i = rows - 1; i >= 1; i--) {          // Lower half of the pattern
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Output –

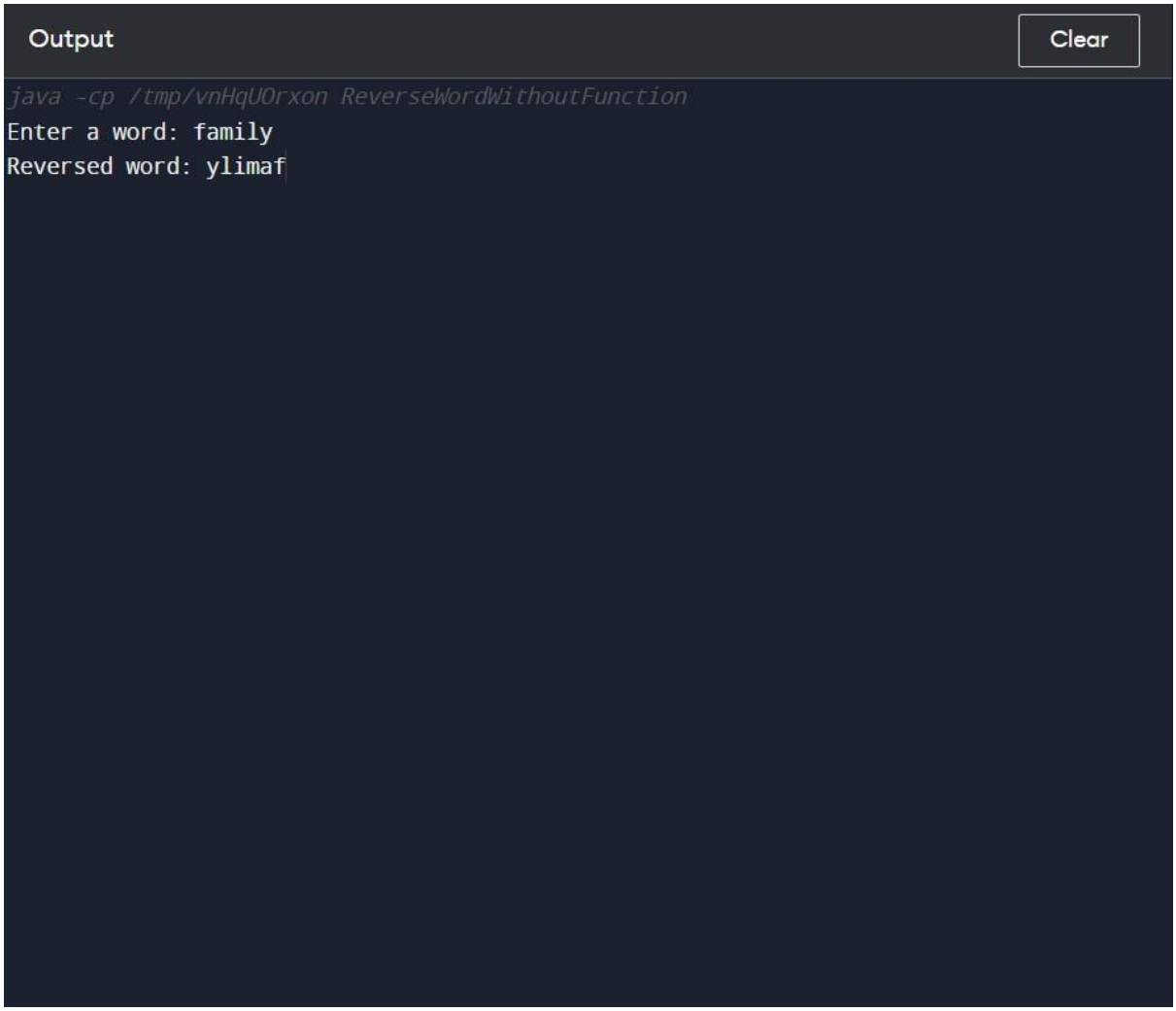


4. Write a JAVA program that accepts a word from the user and reverse it. (should not use any functions)

Code –

```
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 input = scanner.nextLine();
        String reversed = reverseString(input);
        System.out.println("Reversed word: " + reversed);
        scanner.close();
    }

    //Reversing the given string
    public static String reverseString(String str) {
        char[] characters = str.toCharArray();
        int length = characters.length;

        for (int i = 0; i < length / 2; i++) {
            char temp = characters[i];
            characters[i] = characters[length - i - 1];
            characters[length - i - 1] = temp;
        }
        return new String(characters);
    }
}
```

Output –

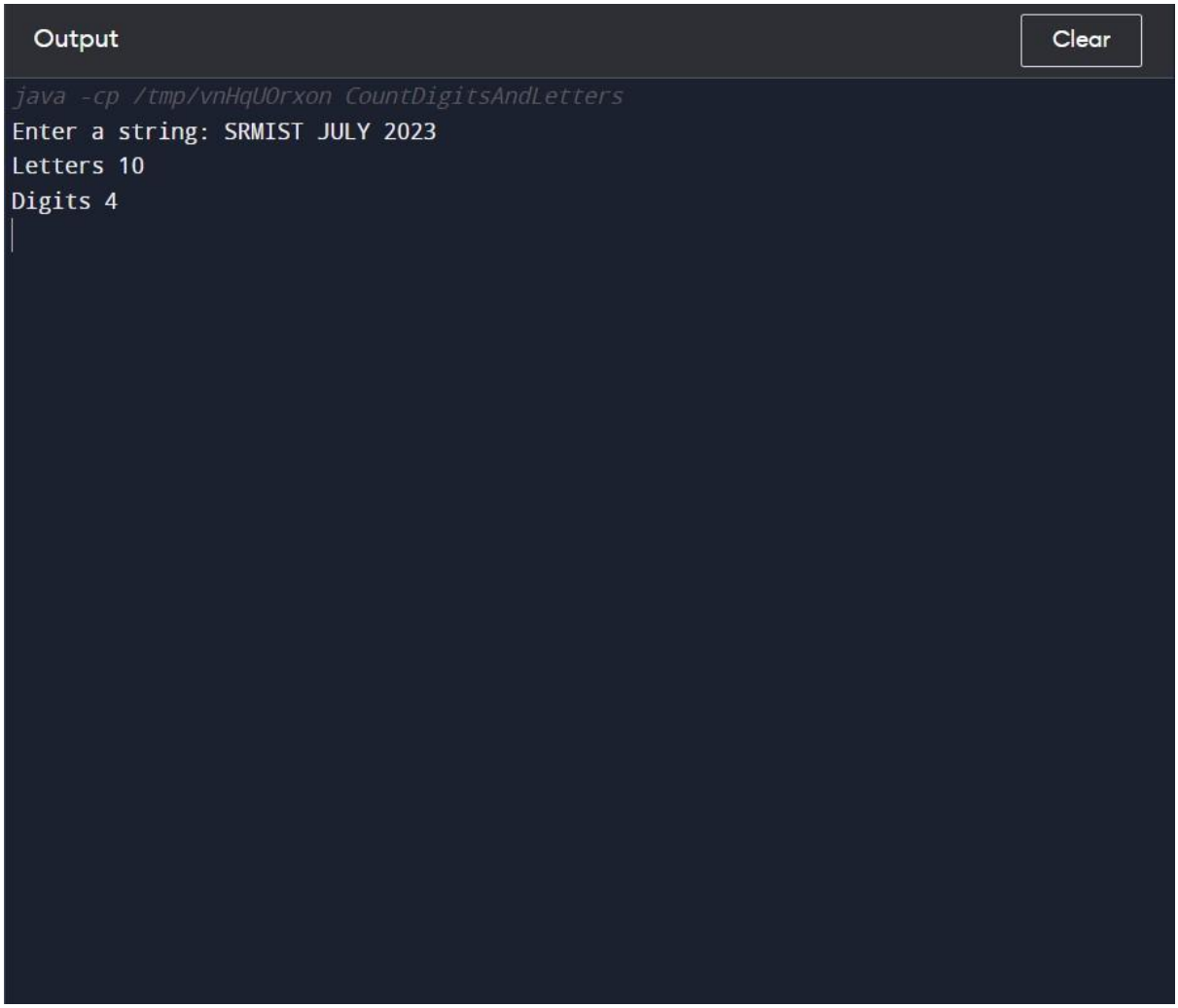

Output
Clear
java -cp /tmp/vnHqUOrxon ReverseWordWithoutFunction
Enter a word: family
Reversed word: ylimaf

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

Code –

```
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 currentChar = input.charAt(i);
            if (Character.isDigit(currentChar)) {
                digitCount++; }
            else if (Character.isLetter(currentChar)) {
                letterCount++; }
        }

        System.out.println("Letters " + letterCount);
        System.out.println("Digits " + digitCount);

        scanner.close();
    }
}
```

Output –


Output
Clear
java -cp /tmp/vnHqUOrxon CountDigitsAndLetters
Enter a string: SRMIST JULY 2023
Letters 10
Digits 4

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

Validation :

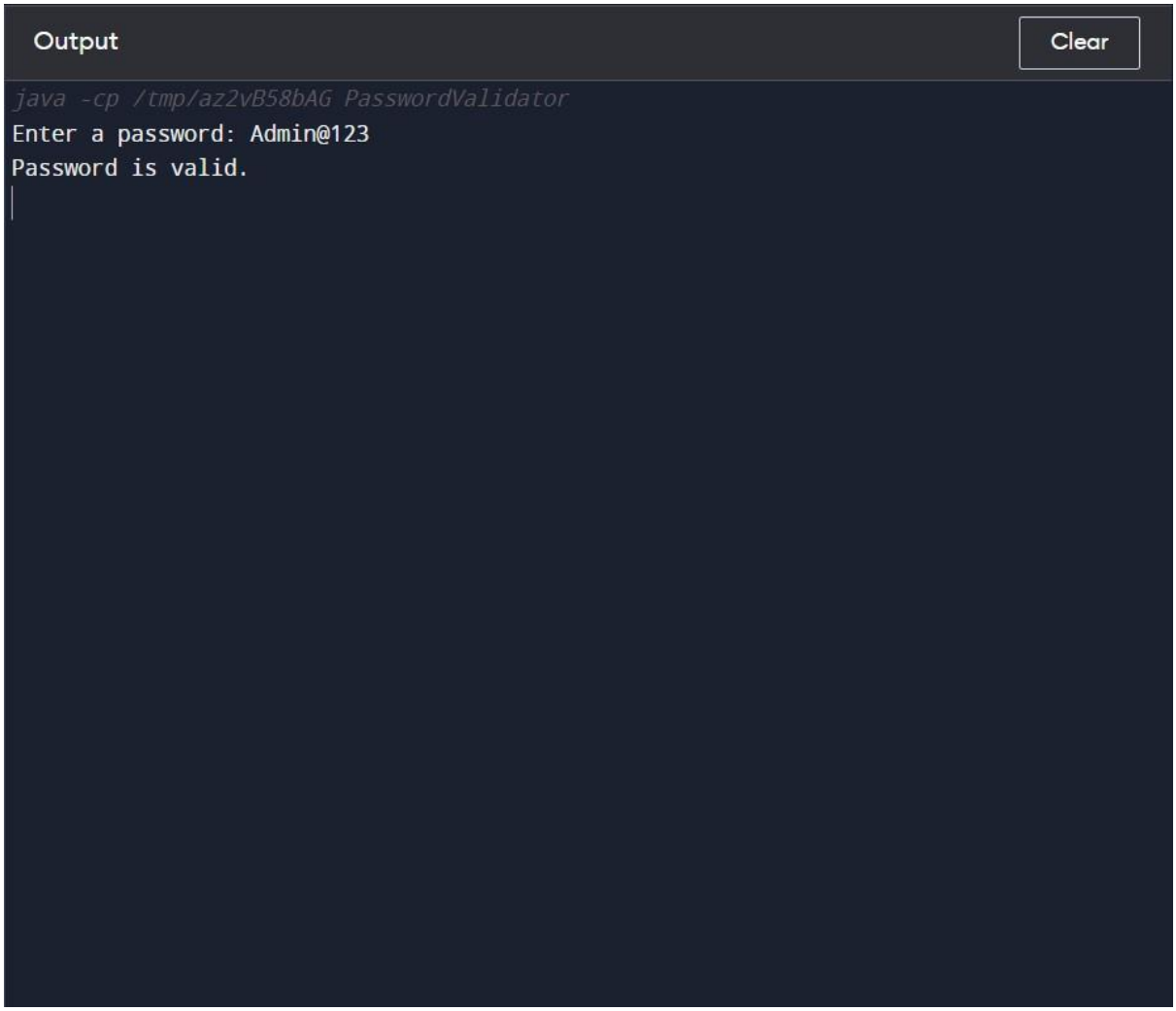
- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

Code –

```
import java.util.Scanner;
public class Password {
   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 not valid."); }
      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 (int i = 0; i < password.length(); i++) {
            char ch = password.charAt(i);

            if (Character.isLowerCase(ch)) {
                hasLower = true;
            } else if (Character.isUpperCase(ch)) {
                hasUpper = true;
            } else if (Character.isDigit(ch)) {
                hasDigit = true;
            } else if (specialChars.contains(Character.toString(ch))) {
                hasSpecial = true;
            }
        }
        return hasLower && hasUpper && hasDigit && hasSpecial;
    }
}
```

Output –

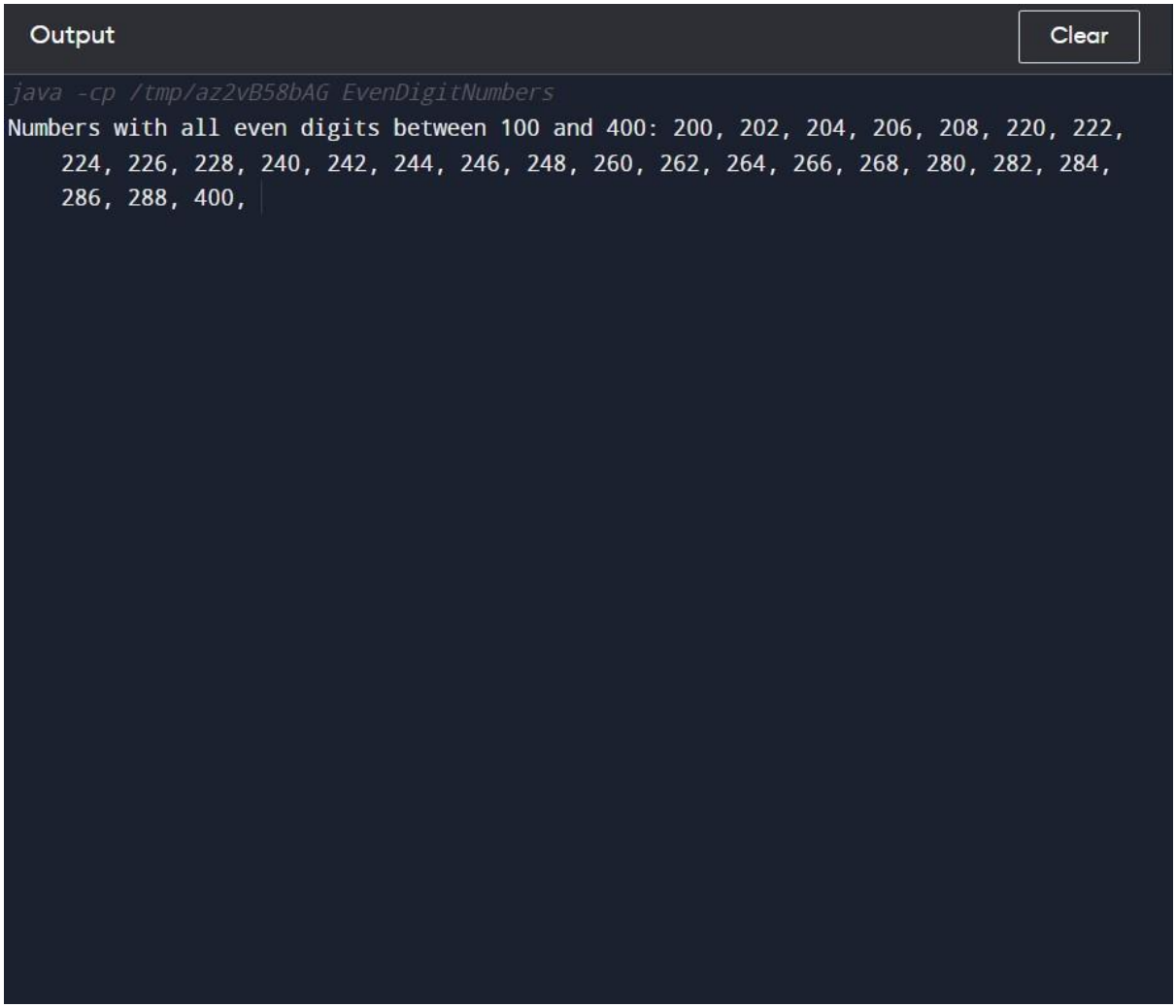

Output
Clear
java -cp /tmp/az2vB58bAG PasswordValidator
Enter a password: Admin@123
Password is valid.

7. 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.print("Numbers with all even digits between 100 and 400: ");
        for (int num = 100; num <= 400; num++) {
            if (hasAllEvenDigits(num)) {
                System.out.print(num + ", ");
            }
        }
    }

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

Output –


Output
Clear
java -cp /tmp/az2vB58bAG EvenDigitNumbers
Numbers with all even digits between 100 and 400: 200, 202, 204, 206, 208, 220, 222,
224, 226, 228, 240, 242, 244, 246, 248, 260, 262, 264, 266, 268, 280, 282, 284,
286, 288, 400,

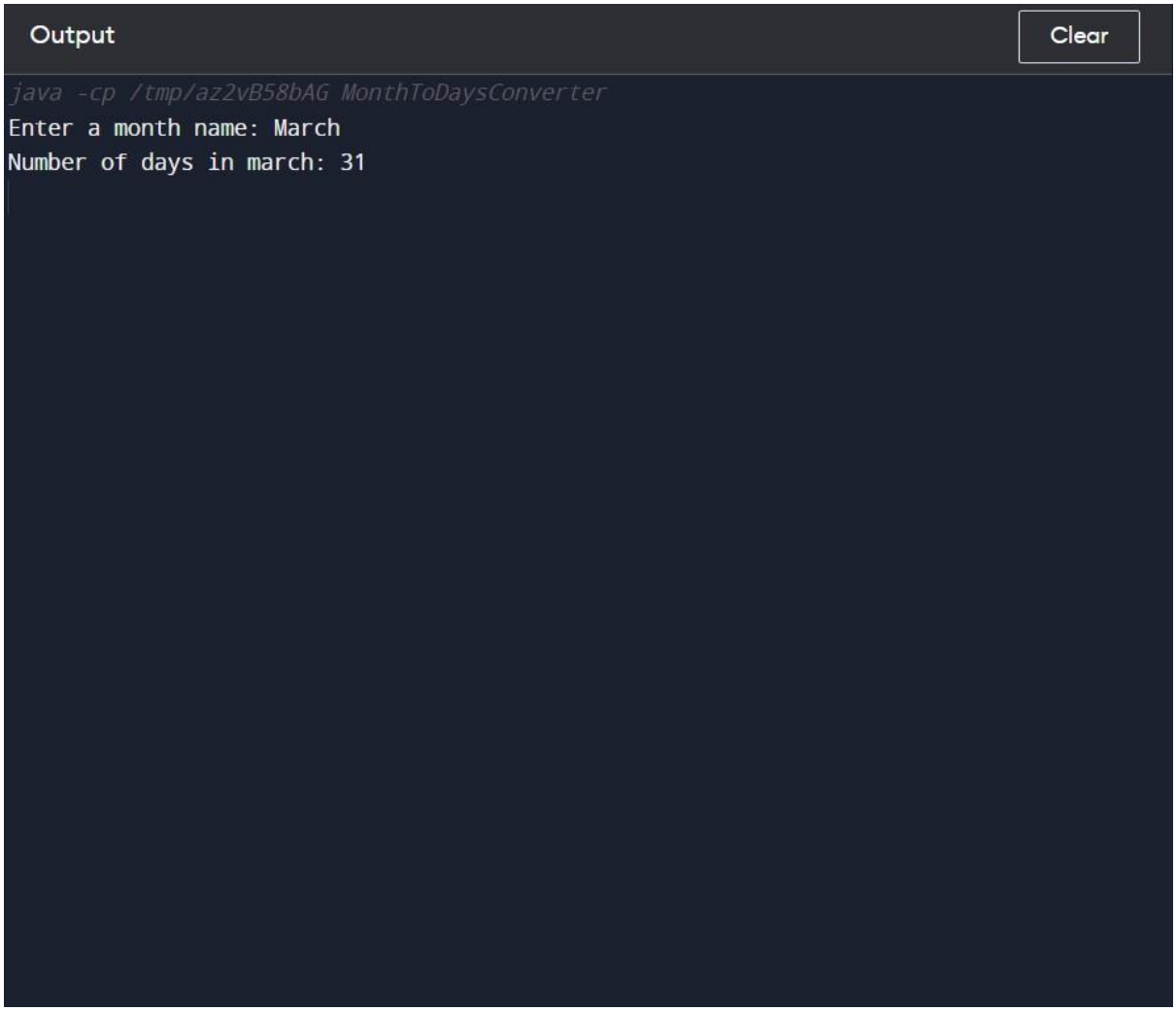
8. Write a JAVA program to convert month name to a number of days.

Code –

```
import java.util.Scanner;
public class MonthToDaysConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a month name: ");
        String monthName = scanner.nextLine().toLowerCase();
        int days;
        switch (monthName) {
            case "january":
            case "march":
            case "may":
            case "july":
            case "august":
            case "october":
            case "december":
                days = 31;
                break;
            case "april":
            case "june":
            case "september":
            case "november":
                days = 30;
                break;
            case "february":
```

```
                days = 28; // Assuming it's not a leap year
                break;
            default:
                days = -1; // Invalid input
                break;
        }
        if (days != -1) {
            System.out.println("Number of days in " + monthName + ": " + days);
        } else {
            System.out.println("Invalid month name.");
        }
    }
}
```

Output –

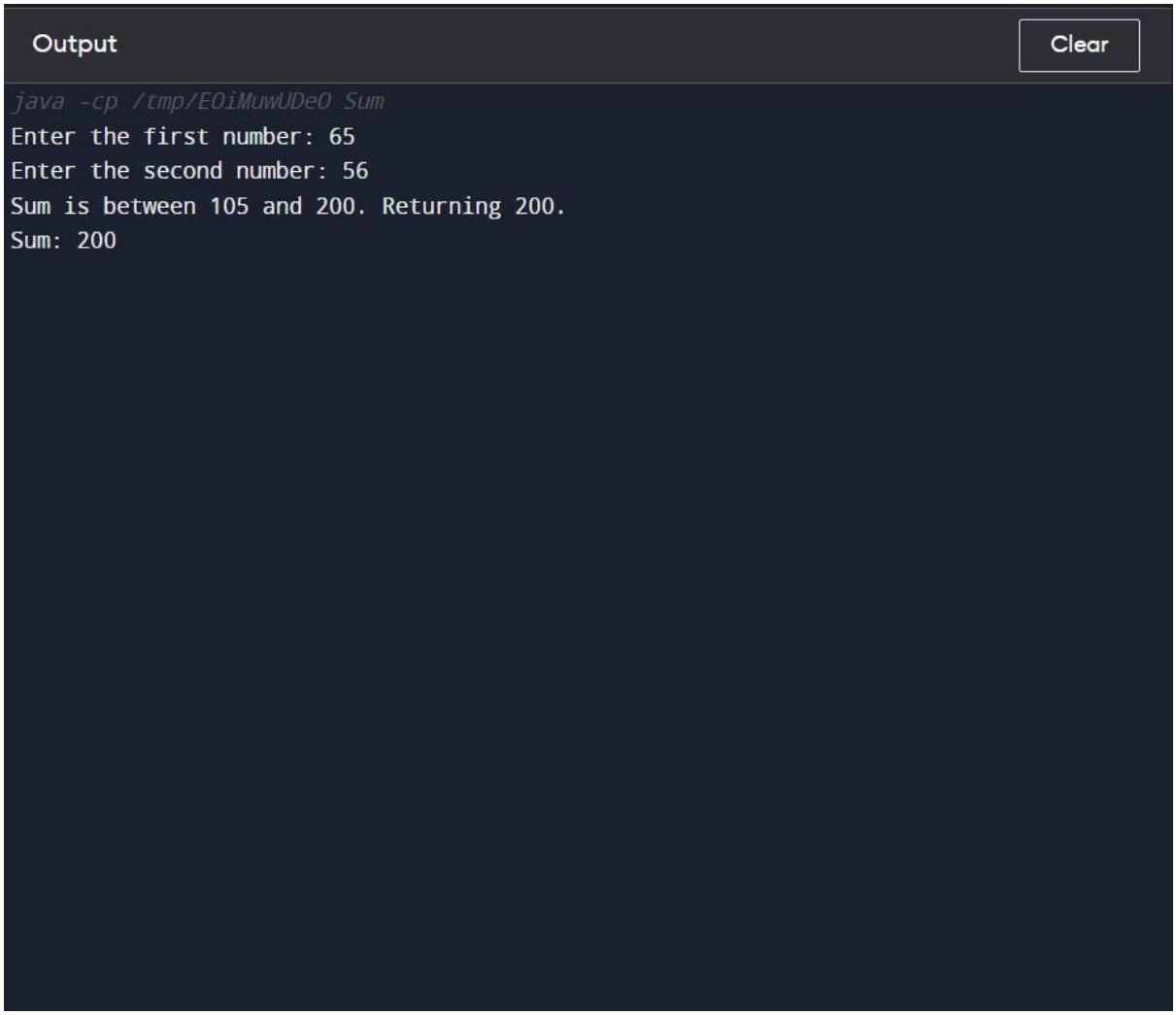

Output
Clear
java -cp /tmp/az2vB58bAG MonthToDaysConverter
Enter a month name: March
Number of days in march: 31

9. 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 Sum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
       //taking input from the user
        System.out.print("Enter the first number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = sc.nextInt();

        int sum = num1 + num2;  //calculating sum

        if (sum >= 105 && sum <= 200) {
            System.out.println("Sum is between 105 and 200. Returning 200.");
            sum = 200;
        }
        System.out.println("Sum: " + sum);
        sc.close();
    }
}
```

Output –


Output
Clear
java -cp /tmp/EOiMuwUDeO Sum
Enter the first number: 65
Enter the second number: 56
Sum is between 105 and 200. Returning 200.
Sum: 200

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

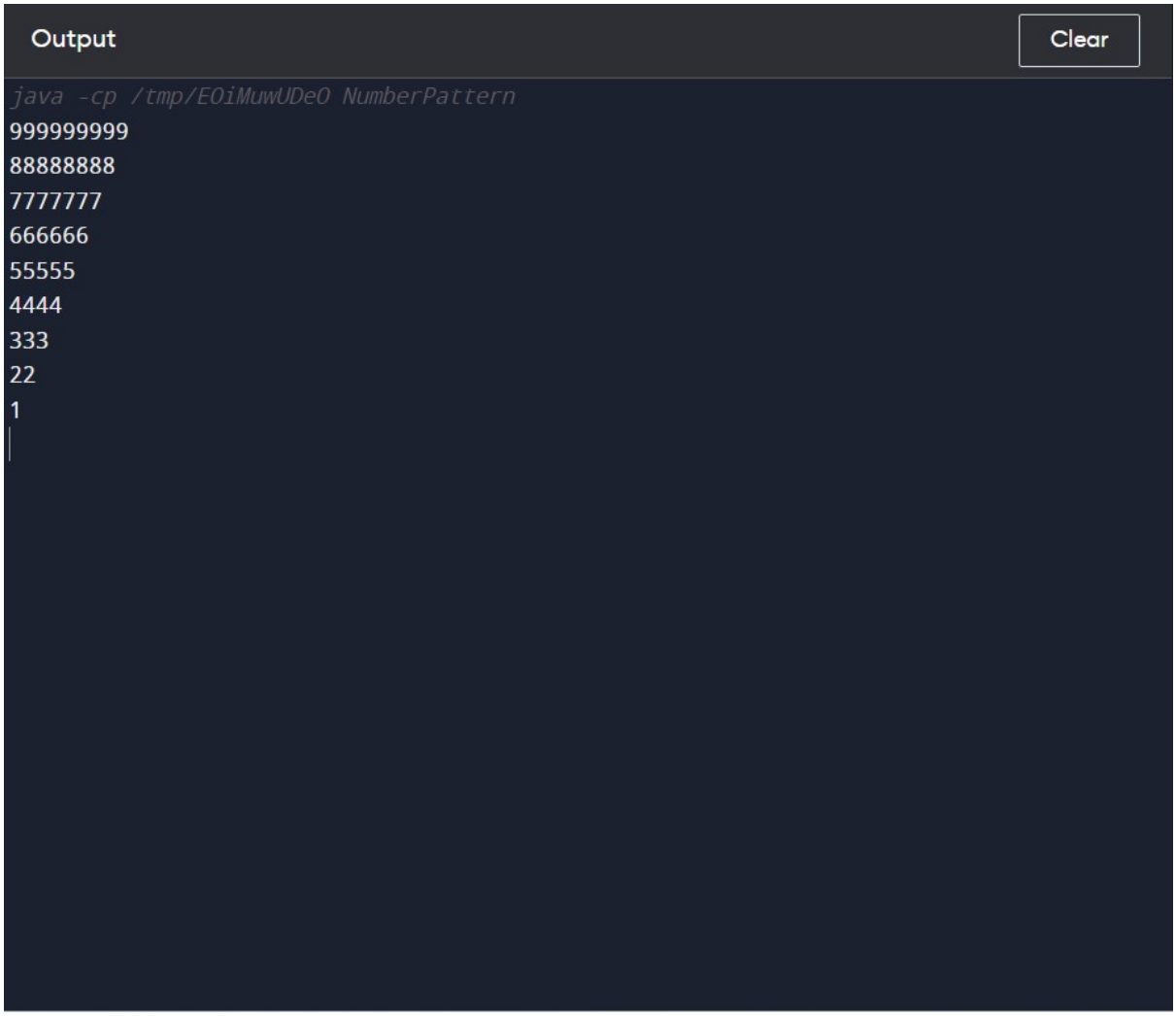
Expected Output:

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

Code –

```
public class NumberPattern {
    public static void main(String[] args) {
        for (int i = 9; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
```

Output –


Output
Clear
java -cp /tmp/EOiMuwUDeO NumberPattern
999999999
88888888
7777777
666666
55555
4444
333
22
1