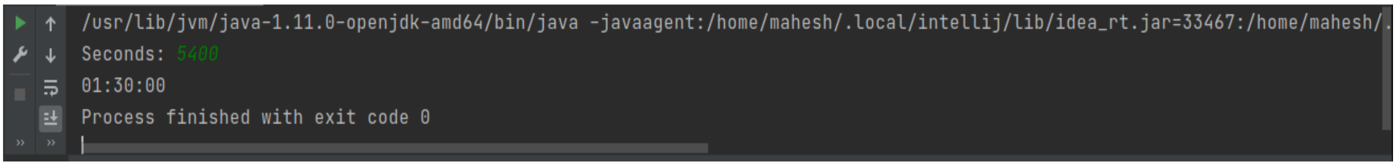
USN: 1RV19CS082

Program 1:

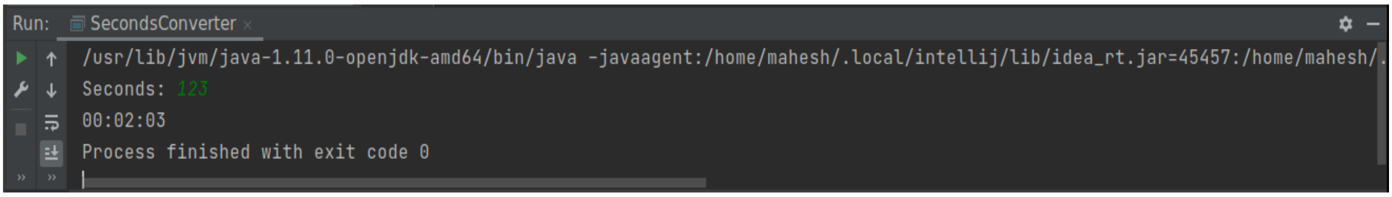
i) Write a Java program to convert time in seconds to hours, minutes and seconds, and display the output in format HH:MM:SS

```
import java.util.Scanner;

public class SecondsConverter {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.print("Seconds: ");

        int total = in.nextInt();

        int seconds = total % 60;

        int minutes = (total / 60) % 60;

        int hours = total / 3600;

        System.out.printf("%02d:%02d:%02d", hours, minutes, seconds);

    }

}
```

Output:



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=33467:/home/mahesh/.
Seconds: 5400
01:30:00
Process finished with exit code 0
```



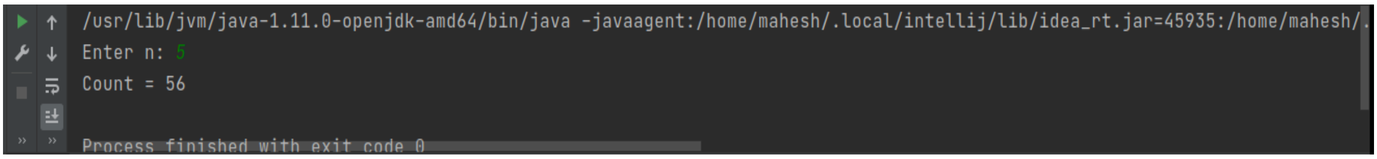
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=45457:/home/mahesh/.
Seconds: 123
00:02:03
Process finished with exit code 0
```

ii) Write a Java program which reads an integer n and find the number of combinations of a,b,c and d ($0 \leq a,b,c,d \leq 9$) where $(a + b + c + d)$ will be equal to n.

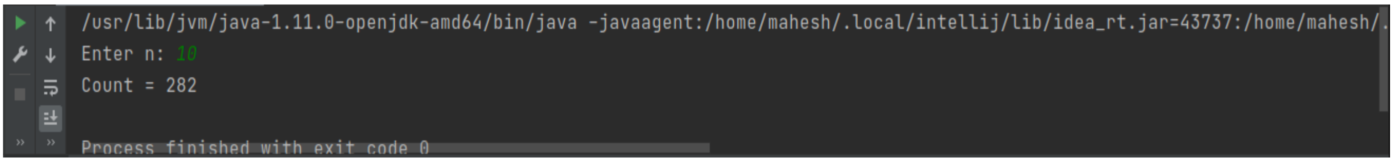
Code:

```
import java.util.Scanner;


public class SumCombinations {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter n: ");
        int count = 0;
        int n = in.nextInt();
        for (int i = 0; i <= 9999; i++) {
            int w = i / 1000; // MSB
            int x = (i / 100) % 10;
            int y = (i / 10) % 10;
            int z = (i % 10);
            if (w+x+y+z == n) {
                count++;
                // System.out.printf("%d + %d + %d + %d\n", w, x, y, z);
            }
        }
        System.out.println("Count = " + count);
    }
}
```

Output:



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=45935:/home/mahesh/.
Enter n: 5
Count = 56

Process finished with exit code 0
```



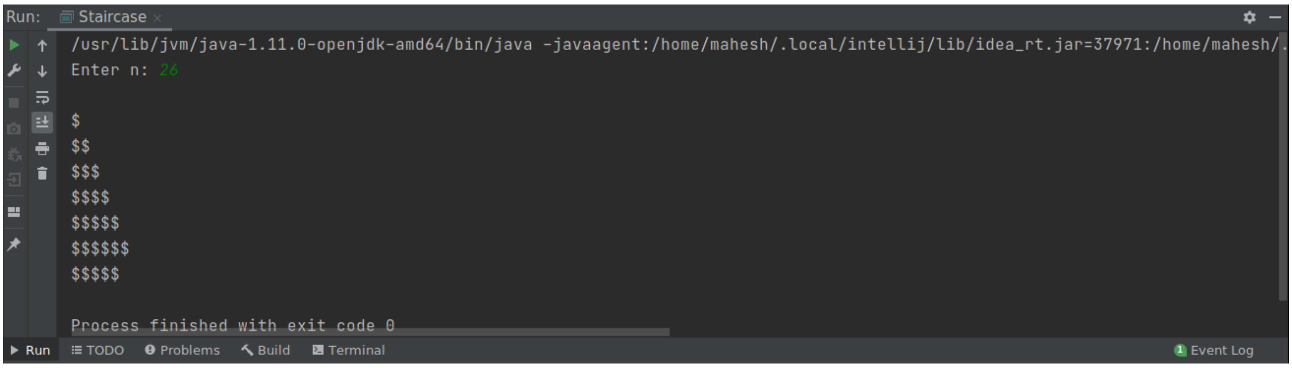
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=43737:/home/mahesh/.
Enter n: 10
Count = 282

Process finished with exit code 0
```

iii) Write a Java program to form a staircase shape of n coins where every k-th row must have exactly k coins. (Put remaining coins in last row)

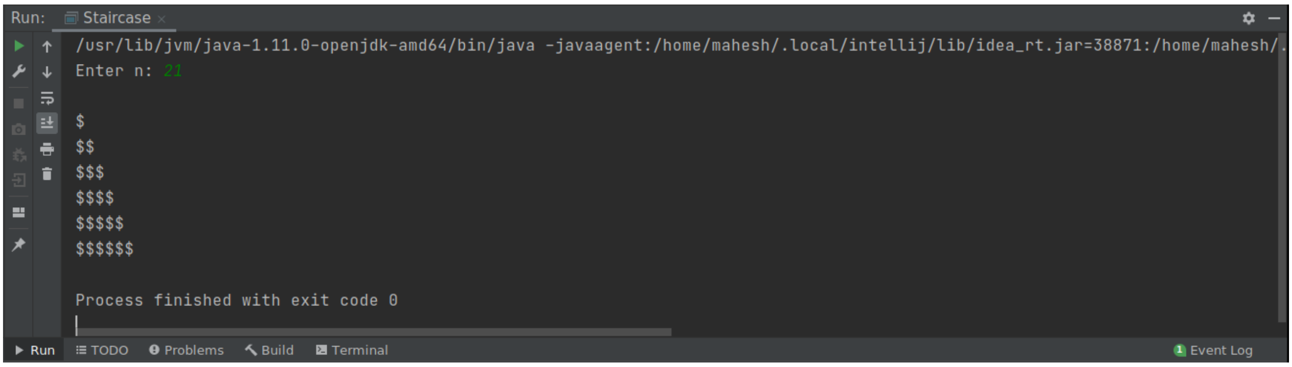
Code:

```
import java.util.Scanner;


public class Staircase {
    static final String pattern = "$";
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter n: ");
        int n = in.nextInt();
        for (int i = 0; true; i++) {
            for (int j = 0; j < i; j++) {
                System.out.print(pattern);
                n--;
                if (n == 0) break;
            }
            System.out.println();
            if (n == 0) break;
        }
    }
}
```

Output:



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=37971:/home/mahesh/.
Enter n: 26

$
$$
$$$
$$$$
$$$$$
$$$$$$
$$$$$

Process finished with exit code 0
```



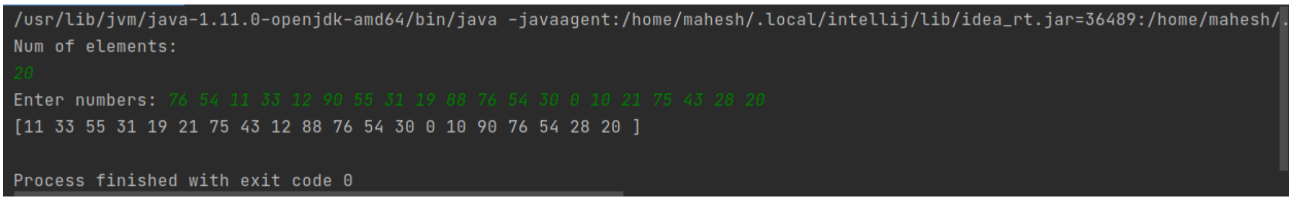
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=38871:/home/mahesh/.
Enter n: 21

$
$$
$$$
$$$$
$$$$$
$$$$$$

Process finished with exit code 0
```

iv) Write a Java program to rearrange all the elements of an given array of integers so that all the odd numbers come before all the even numbers.

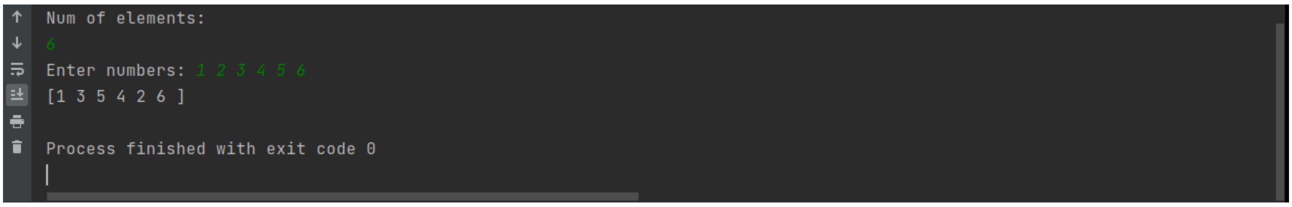
Code:

```
import java.util.Scanner;

public class OddEvenPartition {
    public static void swap(int[] array, int a, int b) {
        int temp = array[a];
        array[a] = array[b];
        array[b] = temp;
    }

    public static int nextOddNum(int[] array, int start) {
        for (int i = start; i < array.length; i++) {
            if (array[i] % 2 == 1) {
                return i;
            }
        }
        return -1;
    }

    public static void partition(int[] array) {
        int i = 0;
        int len = array.length;
        // run through array
        // at any even number, swap with next found odd number
        // and change first_even to that position;
        int oddNumAt = 0;
        while (i < len) {
            oddNumAt = Math.max(i, oddNumAt);
            if (array[i] % 2 != 1) {
                int next = nextOddNum(array, oddNumAt + 1);
                if (next == -1) {
                    return;
                }
                swap(array, i, next);
                oddNumAt = next;
            }
            i++;
        }
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Num of elements: ");
        int len = in.nextInt();
        int[] array = new int[len];
        System.out.print("Enter numbers: ");
        for (int i = 0; i < len; i++) {
            array[i] = in.nextInt();
        }
        partition(array);
        System.out.print("[");
        for (int i = 0; i < len; i++) {
            System.out.printf("%d ", array[i]);
        }
        System.out.println("]");
    }
}
```

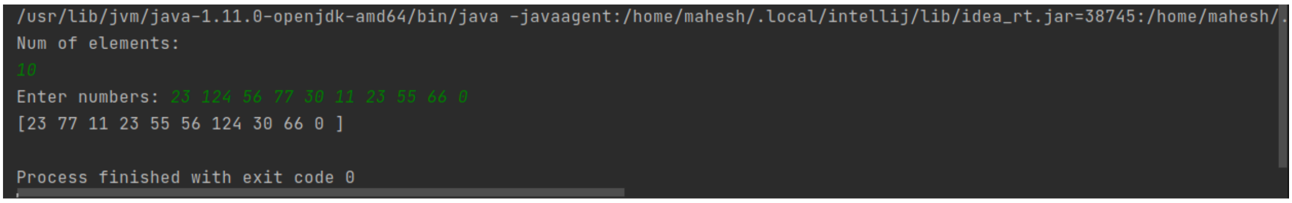
Output:



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=36489:/home/mahesh/.
Num of elements:
20
Enter numbers: 76 54 11 33 12 90 55 31 19 88 76 54 30 0 10 21 75 43 28 20
[11 33 55 31 19 21 75 43 12 88 76 54 30 0 10 90 76 54 28 20 ]

Process finished with exit code 0
```



```
Num of elements:
6
Enter numbers: 1 2 3 4 5 6
[1 3 5 4 2 6 ]

Process finished with exit code 0
```



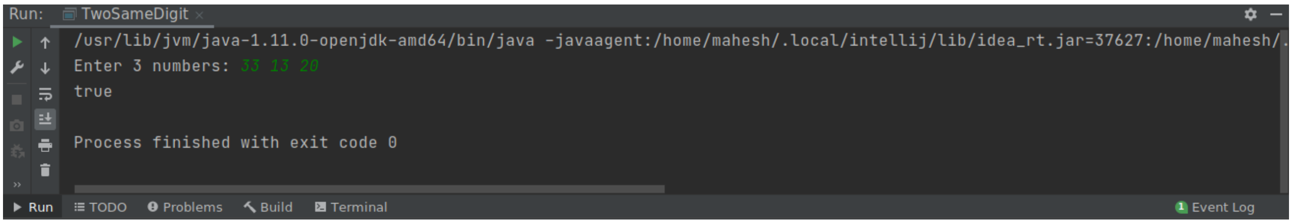
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=38745:/home/mahesh/.
Num of elements:
10
Enter numbers: 23 124 56 77 30 11 23 55 66 0
[23 77 11 23 55 56 124 30 66 0 ]

Process finished with exit code 0
```

v) Write a Java program that accepts three integers from the user and return true if two or more of them (integers) have the same rightmost digit. The integers are non-negative.

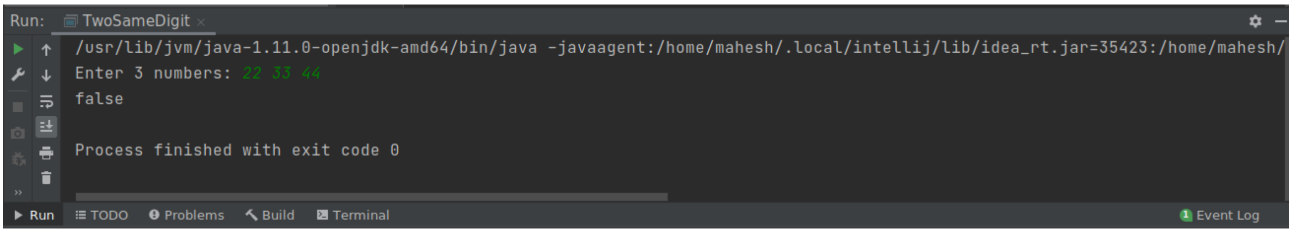
Code:

```
import java.util.Scanner;


public class TwoSameDigit {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter 3 numbers: ");

        int a = in.nextInt() % 10;

        int b = in.nextInt() % 10;

        int c = in.nextInt() % 10;

        System.out.println(a == b || b == c || a == c);

    }
}
```

Output:



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=37627:/home/mahesh/.
Enter 3 numbers: 33 13 20
true

Process finished with exit code 0
```



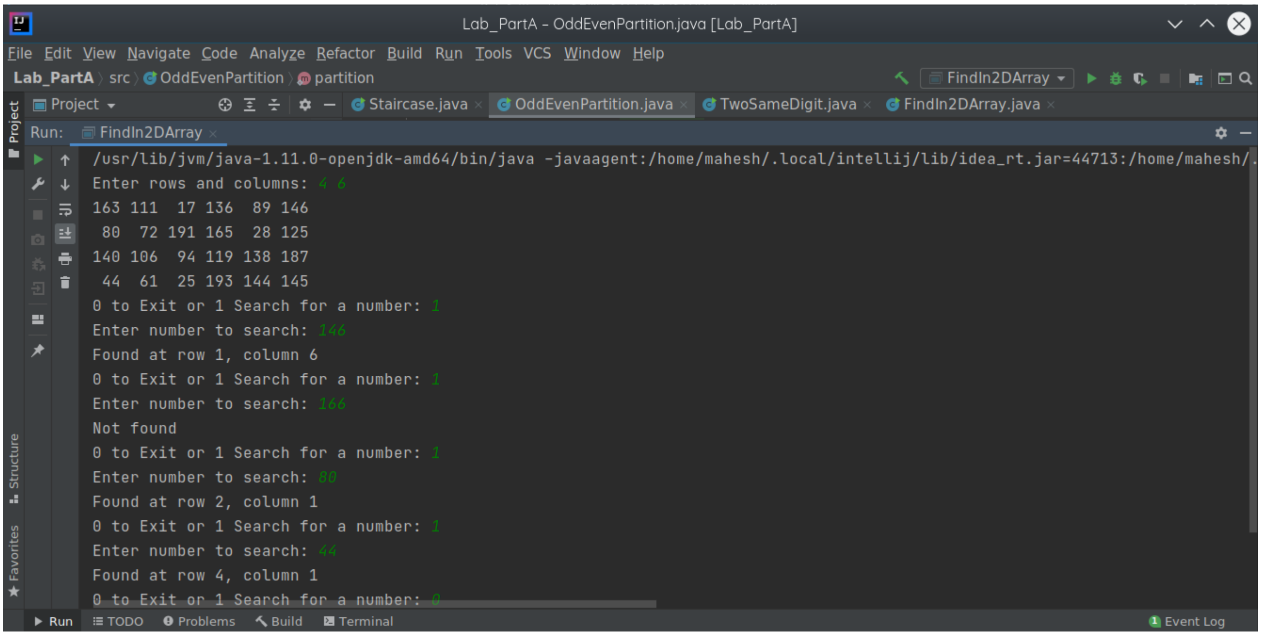
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=35423:/home/mahesh/
Enter 3 numbers: 22 33 44
false

Process finished with exit code 0
```

vi) Given is a 2-dimensional integer array [0..m-1, 0..n-1], each row and column of which is in ascending order (see example) , write a Java program to find the row, column position of a specified number (row, column position) in a given 2-dimensional array.

Code:

```
import java.util.Scanner;


public class FindIn2DArray {

    public static void find_num(int[][] array, int rows, int cols, int
search) {

        for (int i = 0; i < rows; i++) {

            for (int j = 0; j < cols; j++) {

                if (array[i][j] == search) {

                    System.out.printf("Found at row %d, column %d\n",
i+1, j+1);

                    return;

                }

            }

        }

        System.out.println("Not found");

    }
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter rows and columns: ");
        int rows = in.nextInt();
        int cols = in.nextInt();
        int[][] array = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                array[i][j] = (int)(Math.random() * 200);
                // some random number between 0..199
                System.out.printf("%3d ", array[i][j]);
            }
            System.out.println();
        }
        while(true) {
            System.out.print("0 to Exit or 1 Search for a number: ");
            int choice = in.nextInt();
            switch(choice) {
                case 0:
                    System.exit(0);
                case 1:
                    System.out.print("Enter number to search: ");
                    int search = in.nextInt();
                    find_num(array, rows, cols, search);
                    break;
                default:
                    System.out.println("Invalid Option!!");
            }
        }
    }
}
```

Output:



Lab_PartA - OddEvenPartition.java [Lab_PartA]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Lab_PartA src OddEvenPartition partition
FindIn2DArray
Project
Staircase.java
OddEvenPartition.java
TwoSameDigit.java
FindIn2DArray.java
Run:
FindIn2DArray
Structure
Favorites
Run
TODO
Problems
Build
Terminal
Event Log

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/mahesh/.local/intellij/lib/idea_rt.jar=44713:/home/mahesh/.
Enter rows and columns: 4 6
163 111  17 136  89 146
 80  72 191 165  28 125
140 106  94 119 138 187
 44  61  25 193 144 145
0 to Exit or 1 Search for a number: 1
Enter number to search: 146
Found at row 1, column 6
0 to Exit or 1 Search for a number: 1
Enter number to search: 166
Not found
0 to Exit or 1 Search for a number: 1
Enter number to search: 80
Found at row 2, column 1
0 to Exit or 1 Search for a number: 1
Enter number to search: 44
Found at row 4, column 1
0 to Exit or 1 Search for a number: 0
```