

(a)

```

*
**
***
****
*****
*****
*****
*****
*****
*****
*****

```

(b)

```

*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

(c)

```

*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

(d)

```

*
**
***
****
*****
*****
*****
*****
*****
*****

```

Results/Output

Insert pictures for the output of the programs with different inputs

```
[Pythagorean Triples: below 50]
(03, 04, 05)
(05, 12, 13)
(06, 08, 10)
(07, 24, 25)
(08, 15, 17)
(09, 12, 15)
(09, 40, 41)
(10, 24, 26)
(12, 16, 20)
(12, 35, 37)
(14, 48, 50)
(15, 20, 25)
(15, 36, 39)
(16, 30, 34)
(18, 24, 30)
(20, 21, 29)
(21, 28, 35)
(24, 32, 40)
(27, 36, 45)
(30, 40, 50)
```

```
[Pythagorean Triples: below 30]
(03, 04, 05)
(05, 12, 13)
(06, 08, 10)
(07, 24, 25)
(08, 15, 17)
(09, 12, 15)
(10, 24, 26)
(12, 16, 20)
(15, 20, 25)
(18, 24, 30)
(20, 21, 29)
```

```
[Check Palindrome]
12321 : true
55555 : true
45554 : true
11611 : true
16231 : false
46765 : false
```

(a)

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

(b)

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

(c)

[illegible]

(d)

```

          *
        **
       ***
      ****
     *****
    ******
   *******
  ********
 **********
 *

```

Task 2: Create a Point3D class and test it in the Lab01Test class. Point3D class should contain the following data and function members

Data members

x, y, z, 3D point coordinates of type double

Functions members

setCord(double, double, double) : assigns the values for three coordinates

length() : returns distance between point P and origin (0,0,0)

distance(Point3D, Point3D): returns distance between two points

translate(double, double, double)

Results/Output

Insert pictures for the output of the programs with different inputs

```
[Initial Point]
(0.0, 0.0, 0.0)
(0.0, 0.0, 0.0)

[Assigned Point]
(5.7, 3.0, 8.0)
(1.7, 1.0, 3.5)

[The Length Two Points]
Length of p1 Point = 10.270832488167647
Length of p2 Point = 4.017461885320134

[The Distance Between Two Points]
Distance between p1 and p2 = 6.34428877022476

[Translated Point]
(7.2, -3.5, 15.0)
```

Source Code

(1) Lab01Test

```
public class Lab01Test {

    public static void main(String[] args) {
        useFunctions();
        usePoint3D();
    }

    public static void useFunctions() {
        Functions f1 = new Functions();

        f1.getTriples(50);
        f1.getTriples(30);

        System.out.println("[Check Palindrome]");
        f1.isPalindrome(12321); // true
        f1.isPalindrome(55555); // true
        f1.isPalindrome(45554); // true
        f1.isPalindrome(11611); // true
        f1.isPalindrome(16231); // false
        f1.isPalindrome(46765); // false
        System.out.println();

        System.out.println("[Print Triangles]");
        f1.printTriangles(10);
        System.out.println();
    }

    public static void usePoint3D() {
        Point3D p1 = new Point3D();
        Point3D p2 = new Point3D();

        System.out.println("[Initial Point]");
        p1.displayPoint();
        p2.displayPoint();
        System.out.println();

        //System.out.println(p1);
        //System.out.println(p2);

        p1.setCord(5.7, 3.0, 8);
        p2.setCord(1.7, 1.0, 3.5);

        //System.out.println(p1);
        //System.out.println(p2);

        System.out.println("[Assigned Point]");
        p1.displayPoint();
        p2.displayPoint();
        System.out.println();

        System.out.println("[The Length Two Points]");
        System.out.println("Length of p1 Point = " + p1.getLength());
        System.out.println("Length of p2 Point = " + p2.getLength());
        System.out.println();
    }
}
```

```

        System.out.println("[The Distance Between Two Points]");
        System.out.println("Distance between p1 and p2 = " + p1.getDistance(p2));
        System.out.println();

        System.out.println("[Translated Point]");
        p1.translate(1.5, -6.5, 7.0);
        p1.displayPoint();
        System.out.println();
    }
}

```

(2) Functions

```

import java.util.stream.IntStream;

public class Functions {
    public void getTriples(int bound) {
        System.out.println("[Pythagorean Triples: below " + bound + "]");
        IntStream.range(1, bound + 1).forEach(side1 -> IntStream.range(side1, bound + 1)
            .forEach(side2 -> IntStream.range(1, bound + 1).forEach(hypotenuse -> {
                if (side1 * side1 + side2 * side2 == hypotenuse * hypotenuse)
                    System.out.printf("(%02d, %02d, %02d)%n", side1, side2, hypotenuse);
            }
        ));
        System.out.println();
    }

    public boolean isPalindrome(int input) {
        String strNum = Integer.toString(input);
        for (int i = 0; i < strNum.length(); i++) {
            if (strNum.charAt(i) != strNum.charAt(strNum.length() - i - 1)) {
                System.out.println(strNum + " : false");
                return false;
            }
        }
        System.out.println(strNum + " : true");
        return true;
    }

    public void printTriangles(int size) {
        System.out.println("(a)");
        for (int i = 1; i <= size; i++) {
            for (int j = 0; j < i; j++)
                System.out.print("*");
            System.out.println();
        }

        System.out.println("(b)");
        for (int i = size; i >= 1; i--) {
            for (int j = 0; j < i; j++)
                System.out.print("*");
            System.out.println();
        }

        System.out.println("(c)");
        for (int i = size; i >= 1; i--) {
            for (int k = 0; k < size - i; k++)
                System.out.print(" ");
            for (int j = 0; j < i; j++)
                System.out.print("*");
            System.out.println();
        }
    }
}

```

```

    }

    System.out.println("d");
    for (int i = 1; i <= size; i++) {
        for (int k = size - i; k >= 1; k--)
            System.out.print(" ");
        for (int j = 0; j < i; j++)
            System.out.print("*");
        System.out.println();
    }
}

```

(3) Point3D

```

public class Point3D {
    private double x = 0.0;
    private double y = 0.0;
    private double z = 0.0;

    public void displayPoint() {
        System.out.println("(" + x + ", " + y + ", " + z + ")");
    }

    public void setCord(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public double getLength() {
        return Math.sqrt(this.x * this.x + this.y * this.y + this.z * this.z);
    }

    public double getDistance(Point3D p) {
        return Math.sqrt(((x - p.x) * (x - p.x)) + ((y - p.y) * (y - p.y)) + ((z - p.z) * (z - p.z)));
    }

    // public double getDistance(Point3D p1, Point3D p2) {return 0.0;}

    public void translate(double a, double b, double c) {
        this.x += a;
        this.y += b;
        this.z += c;
    }
}

```

Conclusion/Remarks/Feedback

Conclude the Lab. Write your views about it i.e. what have you learned from this lab? It was helpful or difficult/easy etc.

자바로 알고리즘 문제 푸는건 오랜만이라서 재밌었다. 자바가 주 언어가 아니기 때문에 적응하기 어려웠는데, 생각보다 할만 했다. 자바 문법이 좀 어려운 것이 많아서 헛갈리긴 했는데 앞으로 더 열심히 공부해야겠다.

It was fun because it's been a long time since I solved the algorithm problem in Java. It was difficult to adapt because Java was not the main language, but it was easier to solve than I thought. I was confused because there were a lot of difficult Java grammar. I should study harder from now on.