**Name:**\_\_Ulvi Bajarani\_\_\_\_\_\_\_\_\_\_ SID **#:**\_\_20539914\_\_\_\_\_\_\_\_\_

1. Consider the following recursive definition:

mystery ( 1 ) = 1

mystery ( N ) = mystery ( N – 1 ) + 2 \* N – 1

According to above definition.

* 1. What is mystery ( 5 ) ?

mystery ( 5 ) = mystery ( 4 ) + 2 \* 5 – 1. Recursive calls shows that **mystery ( 5 ) equals to 25**.

* 1. For what set of values N is mystery well defined?

**For all Natural N >= 1.**

* 1. For what set of values N is mystery not defined?

**For rational number types (float, double), for all integers N < 1.**

* 1. Complete the method below to implement mystery.

public **static int** mystery ( **int number** )

{

if ( **number == 1** ) //base case;

return **number**;

else // general case

{

return **mystery(number – 1) + 2\*number – 1**;

}

}

1. Consider the following recursive definition:

mystery ( 0, Q ) = Q

mystery ( P, Q ) = mystery ( P – 1, Q + 1 )

According to above definition.

* 1. What is mystery ( 2, 4 ) ?

mystery ( 2, 4 ) = mystery ( 1, 5 ). Recursive calls shows that **mystery ( 2, 4 ) equals to 6**.

* 1. For what set of values P and Q is mystery well defined?

**For all Integer values P >= 0;**

* 1. For what set of values P and Q is mystery not defined?

**For rational number types (float, double), for all integers P < 0.**

* 1. Complete the method below to implement mystery.

public **static int** mystery ( **int P, int Q** )

{

if ( **P == 0** ) //base case;

return Q;

else // general case

{

return **mystery(P-1, Q+1)**;

}

}

1. What does the following method calculate?

public int mystery ( int x, int y )

{

if ( y == 0 ) //base case;

return 0;

else // general case

{

return ( x + mystery(x, y-1) );

}

}

**mystery ( x, 0 ) = 0**

**mystery ( x, y ) = x + mystery ( x, y - 1 )**

1. Design a recursive method to generate a pattern of stars for a given positive integer number *n*.

For instance, if *n* = 4, shall print:

\*\*\*\*

\*\*\*

\*\*

\*

**public static String PrintStars(int row, int stars)**

**{**

**if (row < 1)**

**return "";**

**else if (stars <= row)**

**{**

**System.out.print("\*");**

**return PrintStars(row, stars+1);**

**}**

**else**

**{**

**System.out.println("");**

**return PrintStars(row-1, 1);**

**}**

**}**