

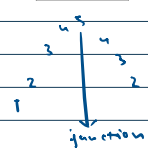
✶ Increasing Star Triangle

* → 1 star
 ** → 2 stars
 *** → 3 stars
 **** → 4 stars
 ***** → 5 stars
 **** → 4 stars
 *** → 3 stars
 ** → 2 stars
 * → 1 star

+1 (stars)

-1 (stars)

n = 5



row < 5 => start
 otherwise : star--

row = 9

row = 2n - 1

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

```
public class Main {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
```

```
        int row = 1;
```

```
        int stars = 1;
```

```
        while(row <= 2*n - 1){
```

```
            int i = 1;
```

```
            while(i <= stars){
```

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

```
                i++;
```

```
            }
```

```
            if(row < n){
```

```
                stars++;
```

```
            }
```

```
            else{
```

```
                stars--;
```

```
            }
```

```
            System.out.println();
```

```
            row++;
```

```
        }
```

```
    }
```

```
}
```

✶ Centered Star Diamond

* → 1
 *** → 3
 ***** → 5
 ***** → 7
 ***** → 9
 ***** → 7
 ***** → 5
 ***** → 3
 * → 1

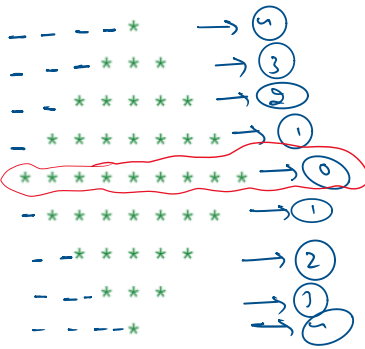
n = 5

stars = stars + 2

stars = stars - 2

junction

☀️ Centered Star Diamond



junction

space --

space + +

if (row < n)
stars ↑

else
stars ↓

space = n - 1

row = 2 * n - 1

stars = 1

stars + 2

stars - 2

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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int row = 1;
        int space = n - 1;
        int star = 1;

        while (row <= 2 * n - 1) {
            System.out.println();
            row++;

            int i = 1;
            while (i <= space) {
                System.out.print("\t");
                i++;
            }

            int j = 1;
            while (j <= star) {
                System.out.print("*\t");
                j++;
            }

            if (row < n) {
                star = star + 2;
                space = space - 1;
            }
            else {
                star = star - 2;
                space = space + 1;
            }
        }
    }
}
```

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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int space = n - 1;
        int star = 1;

        for (int row = 1; row <= 2 * n - 1; row++) {
```

```

for(int j=1; j<=space; j++){
    System.out.print(" ");
}
for(int k=1; k<=star; k++){
    System.out.print("*");
}

if(row < 5){
    star = star + 2;
    space = space - 1;
}
else{
    star = star - 2;
    space = space + 1;
}
System.out.println();
}
}
}

```

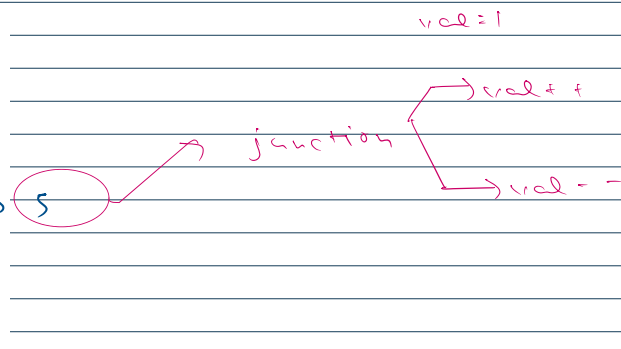
Number Diamond

```

      1
     2 2
    3 3 3
   4 4 4 4
  5 5 5 5 5
   4 4 4 4
    3 3 3
     2 2
      1

```

Arrows indicating the sequence of numbers from 1 to 5 and back down to 1.



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

```

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int row = 1;
        int space = n - 1;
        int star = 1;
        int val = 1;

        while(row <= 2*n-1){
            System.out.println();
            row++;

            int i = 1;
            while(i <= space){
                System.out.print("\t");
                i++;
            }

            int j = 1;
            while(j <= star){
                System.out.print(val + "\t");
                j++;
            }

            if(row < n){
                star = star + 2;
                space = space - 1;
                val++;
            }
            else{

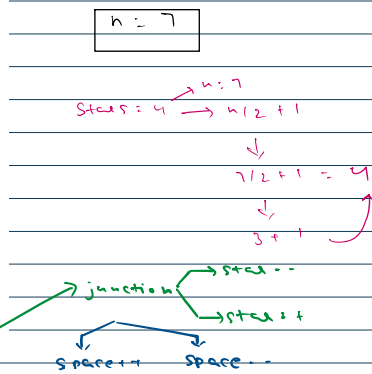
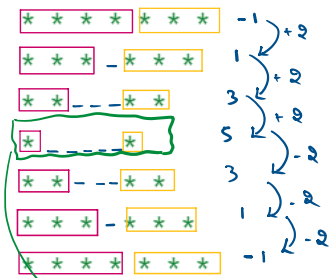
```

```

        star = star - 2;
        space = space + 1;
        val--;
    }
}
}
}

```

★ Hollow Butterfly



```

import java.util.Scanner;

```

```

public class Main {
    public static void main(String[]
args) {
        Scanner sc = new
Scanner(System.in);
        int n = sc.nextInt();

        int row = 1;
        int star = n/2 + 1;
        int space = -1;

        while(row <= n){

            int i = 1;
            while(i <= star){
                System.out.print("* ");
                i++;
            }

            int j = 1;
            while(j <= space){
                System.out.print(" ");
                j++;
            }

            i = 1;
            if(row == 1 || row == n){
                while(i <= star-1){
                    System.out.print("* ");
                    i++;
                }
            }
            else{
                while(i <= star){
                    System.out.print("* ");
                    i++;
                }
            }
        }
    }
}

```

```

if(row <= n/2){
    star--;
}

```

```

        space = space + 2;
    }
    else{
        star++;
        space = space - 2;
    }
    row++;
    System.out.println();
}
}
}

```

Number Pyramid

```

1
2 3 4
5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25
    
```

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int stars = 1;
        int row = 1;
        int spaces = n-1;

        int val = 1;

        while(row <= n){

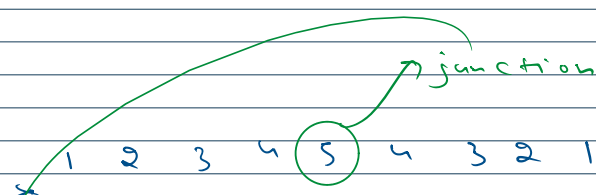
            int k = 1;
            while(k <= spaces){
                System.out.print(" ");
                k++;
            }

            int j = 1;
            while(j <= stars){
                System.out.print(val + " ");
                val++;
                j++;
            }
            System.out.println();
            stars = stars + 2;
            row++;
            spaces--;
        }

        sc.close();
    }
}
    
```

Symmetric Number Pyramid

1

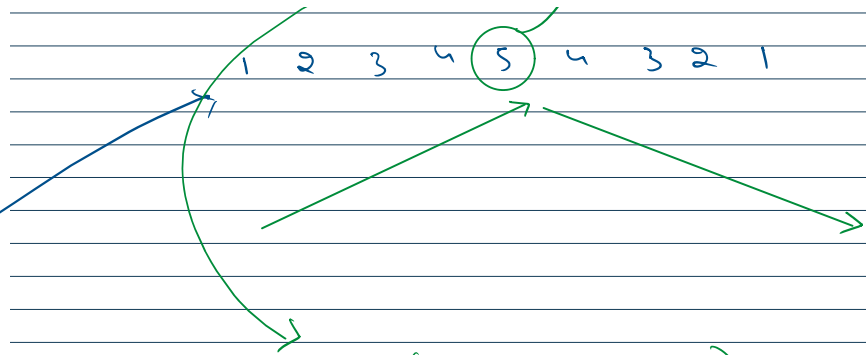


★ Symmetric Number Pyramid

```

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  
```

↓
function



```

if (j <= stars/2)
    val++
else
    val--
  
```

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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
    
```

```
        int stars = 1;
        int row = 1;
        int spaces = n-1;
    
```

```
    while(row <= n){
```

```
        int k = 1;
        while(k <= spaces){
            System.out.print(" ");
            k++;
        }
    
```

```
        int j = 1;
        int val = 1;
        while(j <= stars){
            System.out.print(val + " ");
            if(j <= stars/2){
                val++;
            }
            else{
                val--;
            }
            j++;
        }
    
```

```
        System.out.println();
        stars = stars + 2;
        row++;
        spaces--;
    
```

```
}
```

```
    sc.close();
```

```
}
```

★ Solid Rhombus



`n = 5`

`space = 4`

`start = 5 n`

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int row = 1;
        int space = n-1;
        int star = n;
        while(row <= n){

            int i = 1;
            while(i <= space){
                System.out.print(" ");
                i++;
            }
            space--;

            int k = 1;
            while(k <= star){
                System.out.print("*");
                k++;
            }
            System.out.println();
            row++;
        }

        sc.close();
    }
}
```