

## DamkaBoard

```
1. public class DamkaBoard {
2.     public static void main(String[] args)
3.     {
4.         int n = Integer.parseInt(args[0]);
5.         int i = 0;
6.
7.         while ( i < n)
8.         {
9.             int b = 0;
10.            while (b < n)
11.            {
12.                if (i% 2 != 0 && b == n -1)
13.                {
14.                    System.out.print("*");
15.                }
16.                else{
17.                    System.out.print("* ");
18.                }
19.                b++;
20.            }
21.
22.            if (i%2==0)
23.            {
24.                System.out.println();
25.                System.out.print(" ");
26.            }
27.            else
28.            {
29.                System.out.println();
30.            }
31.
32.            i++;
33.
34.        }
35.    }
36. }
37. }
38.
```

## Divisors

```
1. public class Divisors {
2.     public static void main(String[] args) {
3.         int a = Integer.parseInt(args[0]);
4.         for (int i=1; i <= a ; i ++ )
5.         {
6.             if (a % i == 0 )
7.             {
8.                 System.out.println(i);
9.             }
10.        }
11.    }
12.
13.
14. }
15.
16.
```

## InOrder

```
1. public class InOrder {
2.     public static void main(String[] args)
3.     {
4.         int randnum1 = (int) ((Math.random() * (10-0))+0);
5.         int randnum2 = (int) ((Math.random() * (10-0))+0);
6.         System.out.print(randnum1 + " ");
7.
8.         while ( randnum2 > randnum1)
9.         {
10.             System.out.print(randnum2 + " ");
11.             randnum1= randnum2;
12.             randnum2= (int) ((Math.random() * (10-0))+0);
13.
14.
15.         }
16.     }
17. }
```

## Reverse

```
1. public class Reverse {
2.     public static void main(String[] args)
3.     {
4.         String a = (args[0]);
5.         int len = a.length();
6.         String b = "";
7.         for (int i = len-1; i >= 0 ; i -- )
8.         {
9.             b = b + a.charAt(i);
10.        }
11.
12.        int mid = len / 2 ;
13.
14.        System.out.println(b);
15.        System.out.println("The middle character is " +
16.        b.charAt(mid));
17.
18.    }
19. }
20.
```

Perfect

```
1. public class Perfect {
2.     public static void main(String[] args)
3.     {
4.
5.         int a = Integer.parseInt(args[0]);
6.         String perfect = "";
7.         int sum = 0 ;
8.
9.         for (int i=1; i < a ; i ++ )
10.        {
11.            if (a % i == 0 )
12.            {
13.
14.                if (i != 1)
15.                {
16.                    perfect += " + " + i ;
17.                }
18.                else
19.                {
20.                    perfect += i;
21.                }
22.                sum = sum + i ;
23.
24.            }
25.        }
26.        if (sum == a)
27.        {
28.            System.out.println(a + " is a perfect number since " + a
29.            + " = " + perfect);
30.        }
31.        else
32.        {
33.            System.out.println(a + " is not a perfect number");
34.        }
35.    }
36. }
37.
```

## OneOfEach

```
1. public class OneOfEach {
2.     public static void main(String[] args)
3.     {
4.
5.         boolean g = false ;
6.         boolean b = false ;
7.         int childSum = 0;
8.
9.         double child = (double) ((Math.random() * (1-0))+0);
10.
11.        while ( g == false || b == false)
12.        {
13.
14.            if (child < 0.5)
15.            {
16.                g= true;
17.                System.out.print( " g " );
18.            }
19.
20.            if (child >= 0.5 && child <1)
21.            {
22.                b=true;
23.                System.out.print( " b " );
24.            }
25.            childSum ++;
26.            child = (double) ((Math.random() * (1-0))+0);
27.        }
28.
29.        System.out.println();
30.        System.out.println( "You made it... and you now have "
+ childSum + " children" );
31.
32.
33.
34.    }
35. }
36.
```

## OneOfEachStats1

```
1. public class OneOfEachStats1 {
2.     public static void main(String[] args)
3.     {
4.
5.         double t = Double.parseDouble(args[0]);
6.         int count = 0;
7.         double sumAllFamily = 0;
8.         int twoChild=0;
9.         int threeCchild=0;
10.        int fourChild=0;
11.
12.
13.        boolean g = false ;
14.        boolean b = false ;
15.        int childSum = 0;
16.        double child = (double) ((Math.random() * (1-0))+0);
17.
18.        while (count < t )
19.        {
20.            while ( g == false || b == false)
21.            {
22.
23.                if (child < 0.5)
24.                {
25.                    g= true;
26.                }
27.
28.                if (child >= 0.5 && child <1)
29.                {
30.                    b=true;
31.                }
32.
33.                sumAllFamily ++;
34.                childSum ++;
35.                child = (double) ((Math.random() * (1-0))+0);
36.
37.            }
```

```

38.         if (childSum == 2)
39.         {
40.             twoChild ++;
41.         }
42.         if (childSum == 3)
43.         {
44.             threeCchild++;
45.         }
46.         if (childSum >=4 )
47.         {
48.             fourChild++;
49.         }
50.
51.         childSum = 0;
52.         count ++;
53.         g = false;
54.         b= false ;
55.
56.     }
57.
58.
59.     double average= sumAllFamily/ t ;
60.     int max1 = Math.max(twoChild,threeCchild);
61.     int mostCommonChild = Math.max(max1,fourChild);
62.
63.     System.out.println( "Averag: " + average + "
children to get at least one of each gender" );
64.     System.out.println("Number of families with 2
children: " + twoChild );
65.     System.out.println("Number of families with 3
children: " + threeCchild );
66.     System.out.println("Number of families with 4
children: " + fourChild );
67.     if (mostCommonChild == twoChild)
68.     {
69.         System.out.println("The most common
number of children is 2" );
70.     }
71.
72.     if (mostCommonChild==threeCchild)
73.     {
74.         System.out.println("The most common number
of children is 3" );
75.     }

```



```
76.         if (mostCommonChild == fourChild)
77.             {
78.                 System.out.println("The most common number of children
is 4 or more" );
79.             }
80.         }
81.
```

## OneOfEachStats

```
1. import java.util.Random;
2. public class OneOfEachStats {
3.     public static void main(String[] args)
4.     {
5.
6.         int t = Integer.parseInt(args[0]);
7.         int seed = Integer.parseInt(args[1]);
8.         Random generator = new Random(seed);
9.         int count = 0;
10.        double sumAllFamily = 0;
11.        int twoChild=0;
12.        int threeCchild=0;
13.        int fourChild=0;
14.
15.
16.        boolean g = false ;
17.        boolean b = false ;
18.        int childSum = 0;
19.        double child = (double) (generator.nextDouble());
20.
21.        while (count < t )
22.        {
23.            while ( g == false || b == false)
24.            {
25.
26.                if (child < 0.5)
27.                {
28.                    g= true;
29.                }
30.
31.                if (child >= 0.5 && child <1)
32.                {
33.                    b=true;
34.                }
35.
36.                sumAllFamily ++;
37.                childSum ++;
38.                child = (double) (generator.nextDouble());
39.
40.            }
41.            if (childSum == 2)
42.            {
43.                twoChild ++;
44.            }
45.            if (childSum == 3)
```

```

46.         {
47.             threeCchild++;
48.         }
49.         if (childSum >=4 )
50.         {
51.             fourChild++;
52.         }
53.
54.         childSum = 0;
55.         count ++;
56.         g = false;
57.         b= false ;
58.
59.     }
60.
61.
62.         double average= sumAllFamily/ t ;
63.         int max1 = Math.max(twoChild,threeCchild);
64.         int mostCommonChild = Math.max(max1,fourChild);
65.
66.         System.out.println("Average: " + average + "
children to get at least one of each gender.");
67.         System.out.println("Number of families with 2
children: " + twoChild);
68.         System.out.println("Number of families with 3
children: " + threeCchild);
69.         System.out.println("Number of families with 4 or
more children: " + fourChild);
70.         if (mostCommonChild == twoChild)
71.         {
72.             System.out.println("The most common
number of children is 2.");
73.         }
74.
75.         if (mostCommonChild==threeCchild)
76.         {
77.             System.out.println("The most common number
of children is 3.");
78.         }
79.         if (mostCommonChild == fourChild)
80.             System.out.println("The most common number of children
is 4 or more.");
81.         }
82.     }
83. }

```

