Problems on loops-2

Assignment Solutions

SKILLS

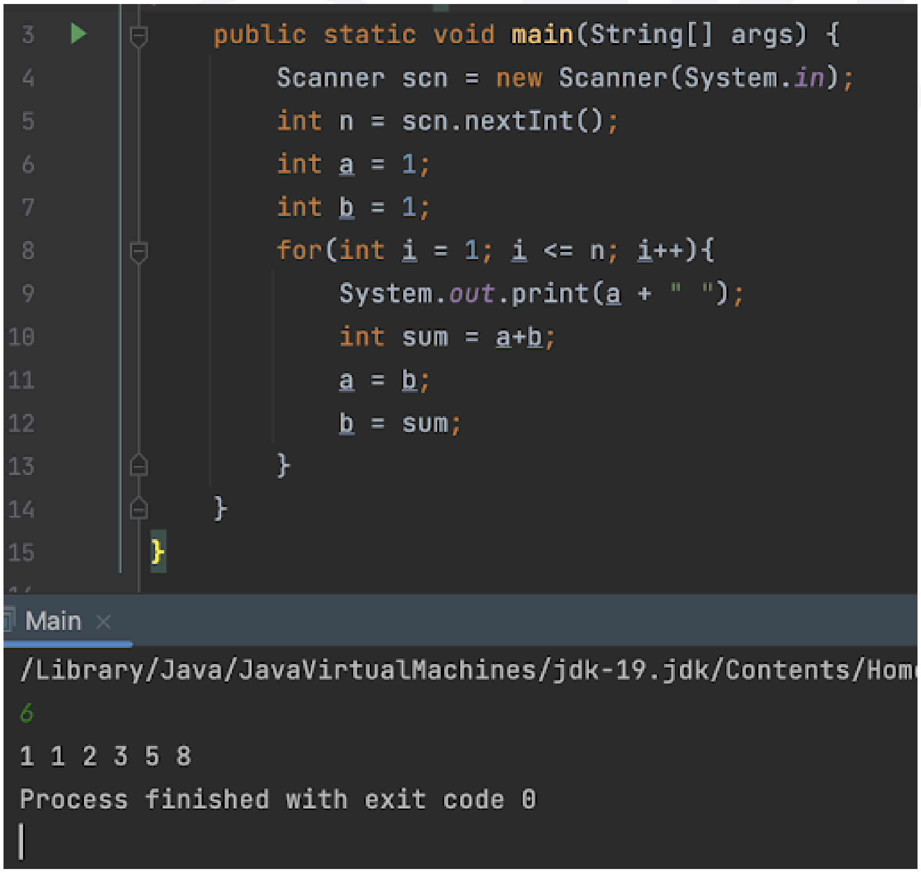
Q1 - Write a program to print Fibonacci series of n terms where n is input by user.

Input:
6

Expected Output:
1 1 2 3 5 8

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int a = 1;  //initial 2 elements are 1 and 1
    int b = 1;
    for(int i = 1; i <= n; i++){
        System.out.print(a + “ ”);
        int sum  = a+b; //calculating every 3rd element in the series by summing up previous 2
        a = b; //update previous element to next element
        b = sum;//update b to newly created next element
    }
   }
}
```



```
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int a = 1;
        int b = 1;
        for(int i = 1; i <= n; i++){
            System.out.print(a + " ");
            int sum = a+b;
            a = b;
            b = sum;
        }
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Hom
6
1 1 2 3 5 8
Process finished with exit code 0
```

Q2 - Write a program to enter the numbers till the user wants, the number can be positive, negative or zero. Calculate the sum of numbers until a negative number is encountered. I f the input is a negative number, current sum is discarded and print -1.

Input:

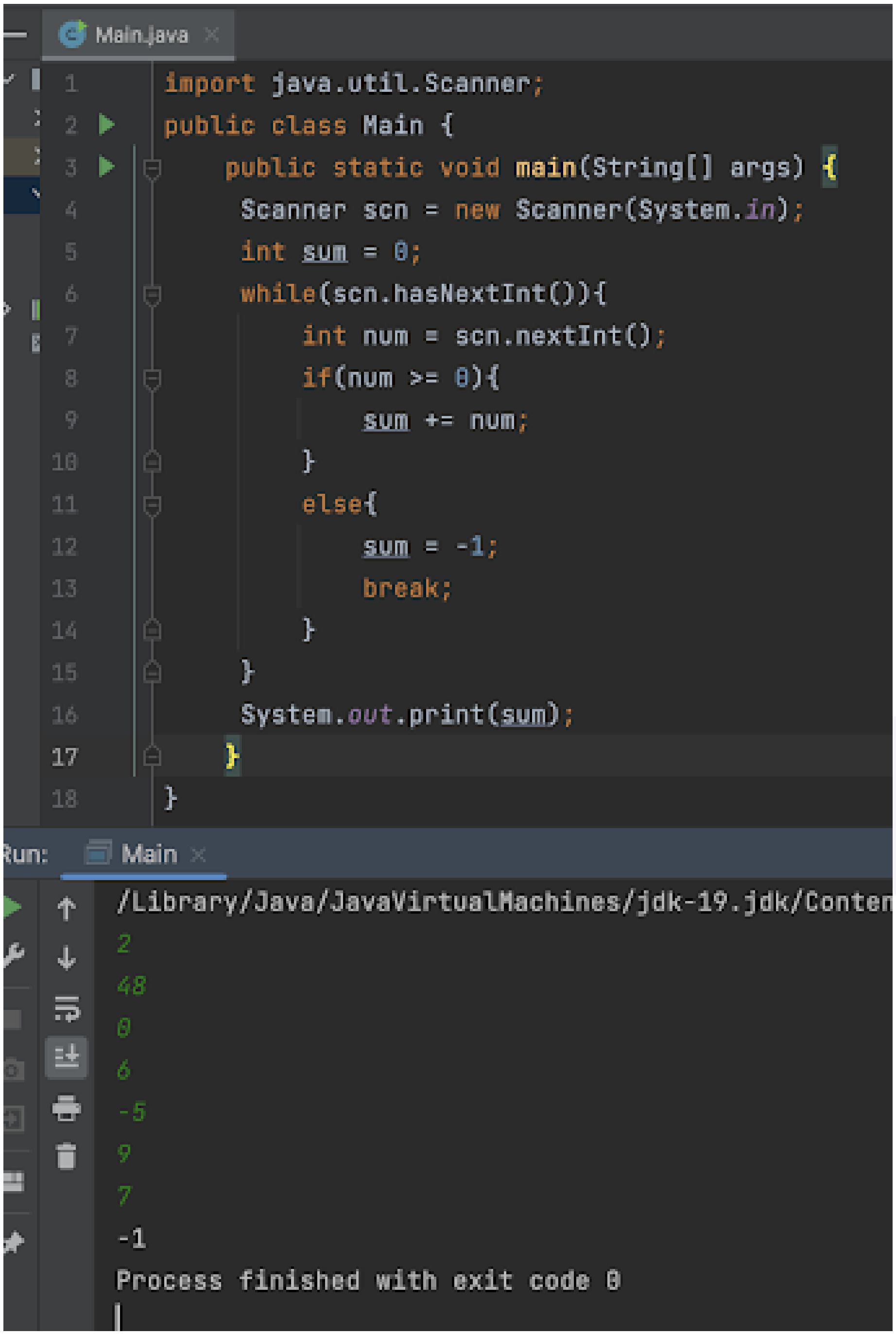
2
48
0
6
-5
9
7

Expected Output1:

-1

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int sum = 0;
    while(scn.hasNextInt()){ //check if input exists or not
        int num = scn.nextInt();
        if(num >= 0){ //if input is positive or zero add it to the current sum
            sum += num;
        }
        else{
            sum = -1;//input is negative so change sum to -1 and break out of the loop
            break;
        }
    }
    System.out.print(sum);
   }
}
```



```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
     Scanner scn = new Scanner(System.in);
     int sum = 0;
     while(scn.hasNextInt()){
         int num = scn.nextInt();
         if(num >= 0){
             sum += num;
         }
         else{
             sum = -1;
             break;
         }
     }
     System.out.print(sum);
    }
}
```

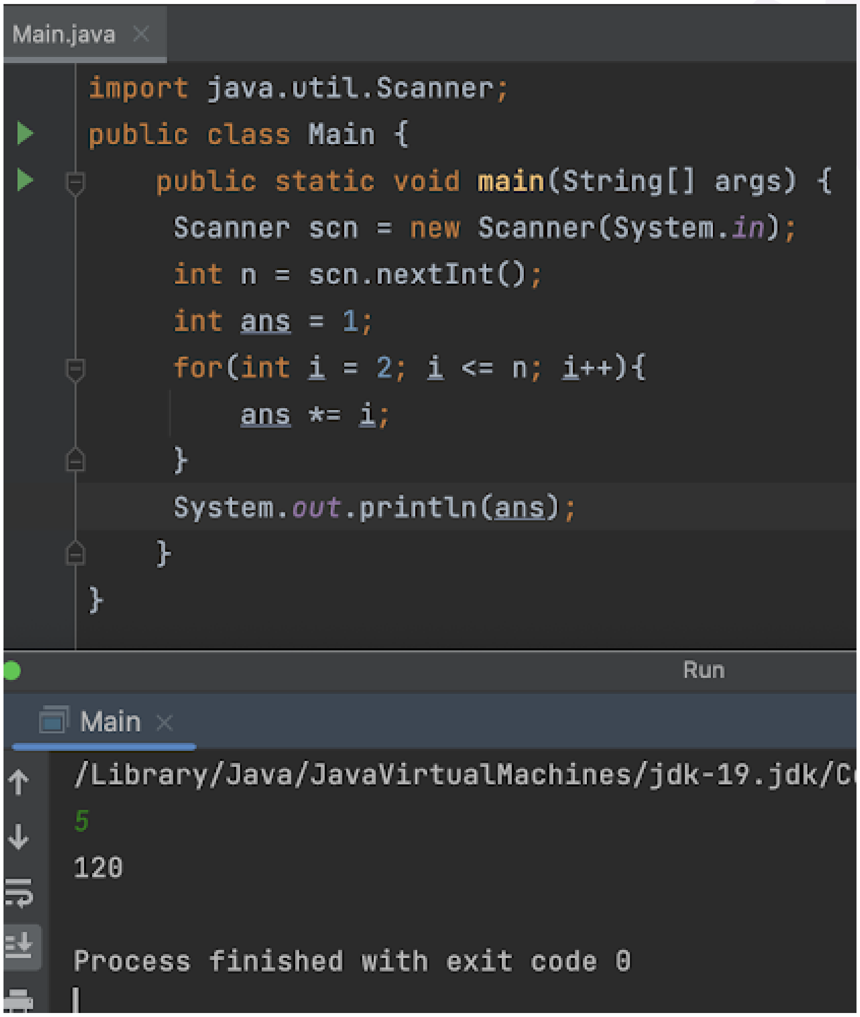
```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Conten
2
48
0
6
-5
9
7
-1
Process finished with exit code 0
```

Q3- Write a program to calculate the factorial of a number.

Input:
5
Expected Output:
120

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int ans = 1; //initialize with 1 as factorial of 0 and 1 both is 1
    for(int i = 2; i <= n; i++){
        ans *= i; //keep multiplying numbers in the current product till you reach n
    }
    System.out.println(ans);
   }
}
```



```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
     Scanner scn = new Scanner(System.in);
     int n = scn.nextInt();
     int ans = 1;
     for(int i = 2; i <= n; i++){
         ans *= i;
     }
     System.out.println(ans);
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/C
5
120

Process finished with exit code 0
```

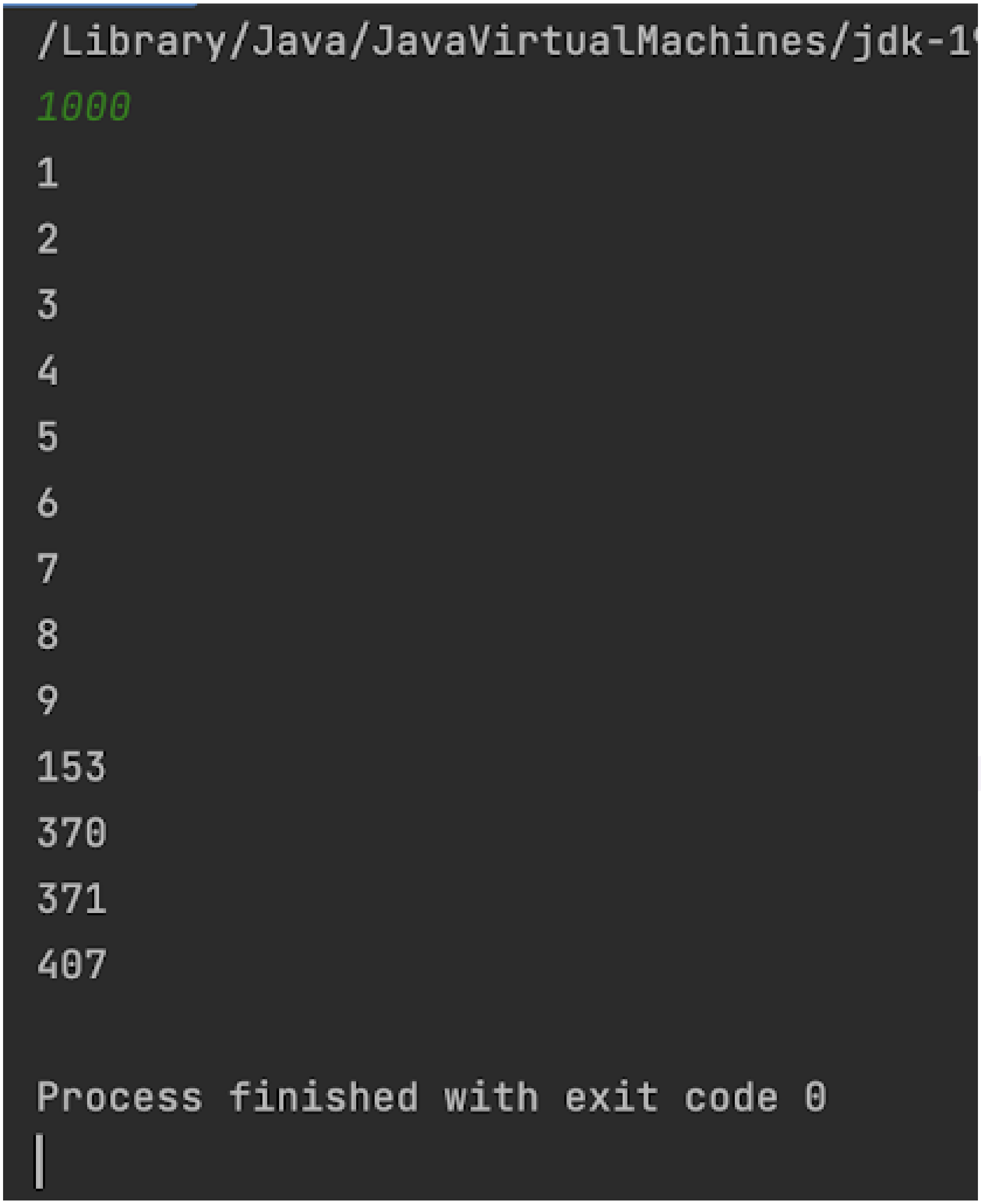
Q4- Write a program to print all Armstrong numbers between 1 to n.

Input:
1000

Expected Output:
153
370
371
407

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);
       int n = scn.nextInt();
       int num = 1; //variable that will check for each number starting from 1 till n
       while(num <= n){
           int count = 0;
           int i = num;
           while(i > 0){
               count++;
               i /= 10;
           }
           int val = num; //store value of current num in a temporary variable to calculate sum
           of its digits
           int sum = 0;
           while(val > 0){ //break the number digit by digit until it reaches 0
               int digit = val % 10;
               sum += Math.pow(digit, count);//add cube of digit to sum
               val /= 10;
               if(sum > val){ //if sum has exceeded our current number, there is no way for it
               to be an Armstrong number
                   continue;
               }
           }
           if(sum == num){ //this is an armstrong number
               System.out.println(num);

           }
           num++;
       }
   }
```



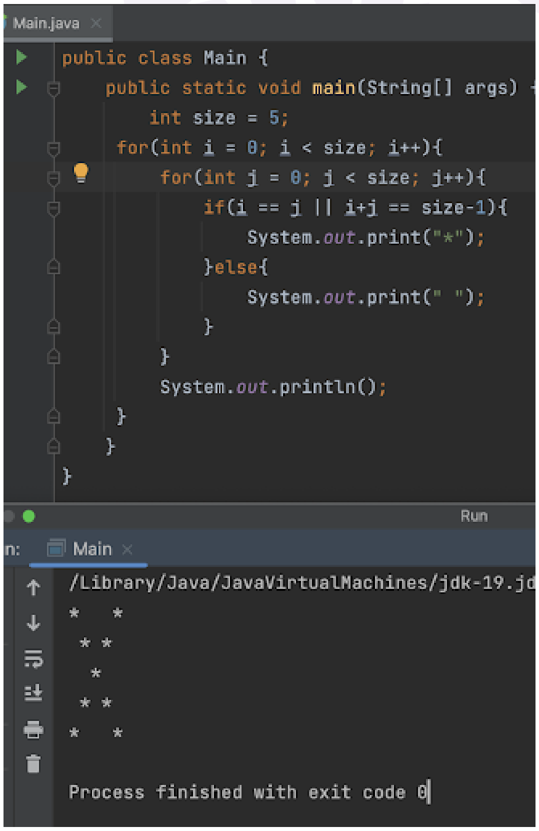
```
/Library/Java/JavaVirtualMachines/jdk-1
1000
1
2
3
4
5
6
7
8
9
153
370
371
407

Process finished with exit code 0
```

Q5 – Write a program to print the cross pattern given below (in the shape of X):

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

Code:

```
public class Test {
   public static void main(String[] args) {
       int size = 5; //length of pattern
    for(int i = 0; i < size; i++){
     for(int j = 0; j < size; j++){
     if(i == j || i+j == size-1){ //consider a rectangle, we need to print stars where both
     diagonals are present
                System.out.print("*");
            }else{
                System.out.print(" ");
            }
        }
        System.out.println();
    }
   }
}
```



Main.java

```
public class Main {
    public static void main(String[] args) {
        int size = 5;
     for(int i = 0; i < size; i++){
         for(int j = 0; j < size; j++){
             if(i == j || i+j == size-1){
                 System.out.print("*");
             }else{
                 System.out.print(" ");
             }
         }
         System.out.println();
     }
    }
}
```

Run

Main

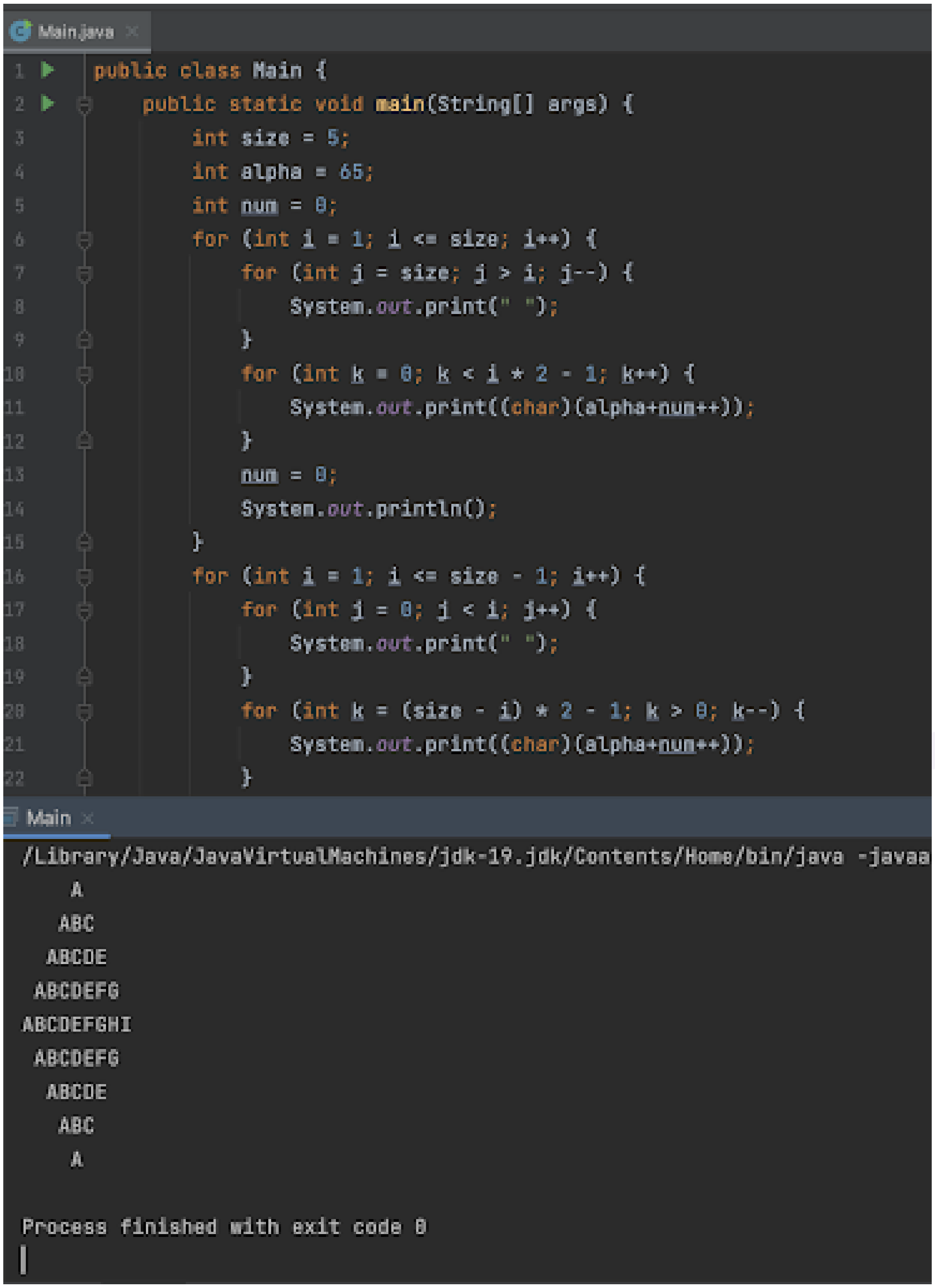
```
/Library/Java/JavaVirtualMachines/jdk-19.jd
*   *
 * *
  *
 * *
*   *

Process finished with exit code 0
```

Q6- Write a program to print alphabet diamond pattern:

```
    A
   ABC
  ABCDE
 ABCDEFG
ABCDEFGHI
 ABCDEFG
  ABCDE
   ABC
    A
```

Code:

```java
public class Test {
   public static void main(String[] args) {
       int size = 5; //length of pattern is 5 only, the below part is the upside down version of top part with 5th line in the middle, with total of (2n-1) lines
       int alpha = 65; //ASCII code for first capital letter
       int num = 0; //will increment alpha as char progresses
       for (int i = 1; i <= size; i++) {
           for (int j = size; j > i; j--) {
               System.out.print(" "); //top half until the spaces need to be printed
           }
           for (int k = 0; k < i * 2 - 1; k++) {
               System.out.print((char)(alpha+num++));//top half until the char need to be printed, next char can be achieved by incrementing the ASCII code by 1
           }
           num = 0;
           System.out.println();
       }
       for (int i = 1; i <= size - 1; i++) {
           for (int j = 0; j < i; j++) { //bottom half
               System.out.print(" ");
           }
           for (int k = (size - i) * 2 - 1; k > 0; k--) {
               System.out.print((char)(alpha+num++));
           }
           num = 0;
           System.out.println();
       }
   }
}
```



```
public class Main {
    public static void main(String[] args) {
        int size = 5;
        int alpha = 65;
        int num = 0;
        for (int i = 1; i <= size; i++) {
            for (int j = size; j > i; j--) {
                System.out.print(" ");
            }
            for (int k = 0; k < i * 2 - 1; k++) {
                System.out.print((char)(alpha+num++));
            }
            num = 0;
            System.out.println();
        }
        for (int i = 1; i <= size - 1; i++) {
            for (int j = 0; j < i; j++) {
                System.out.print(" ");
            }
            for (int k = (size - i) * 2 - 1; k > 0; k--) {
                System.out.print((char)(alpha+num++));
            }
```

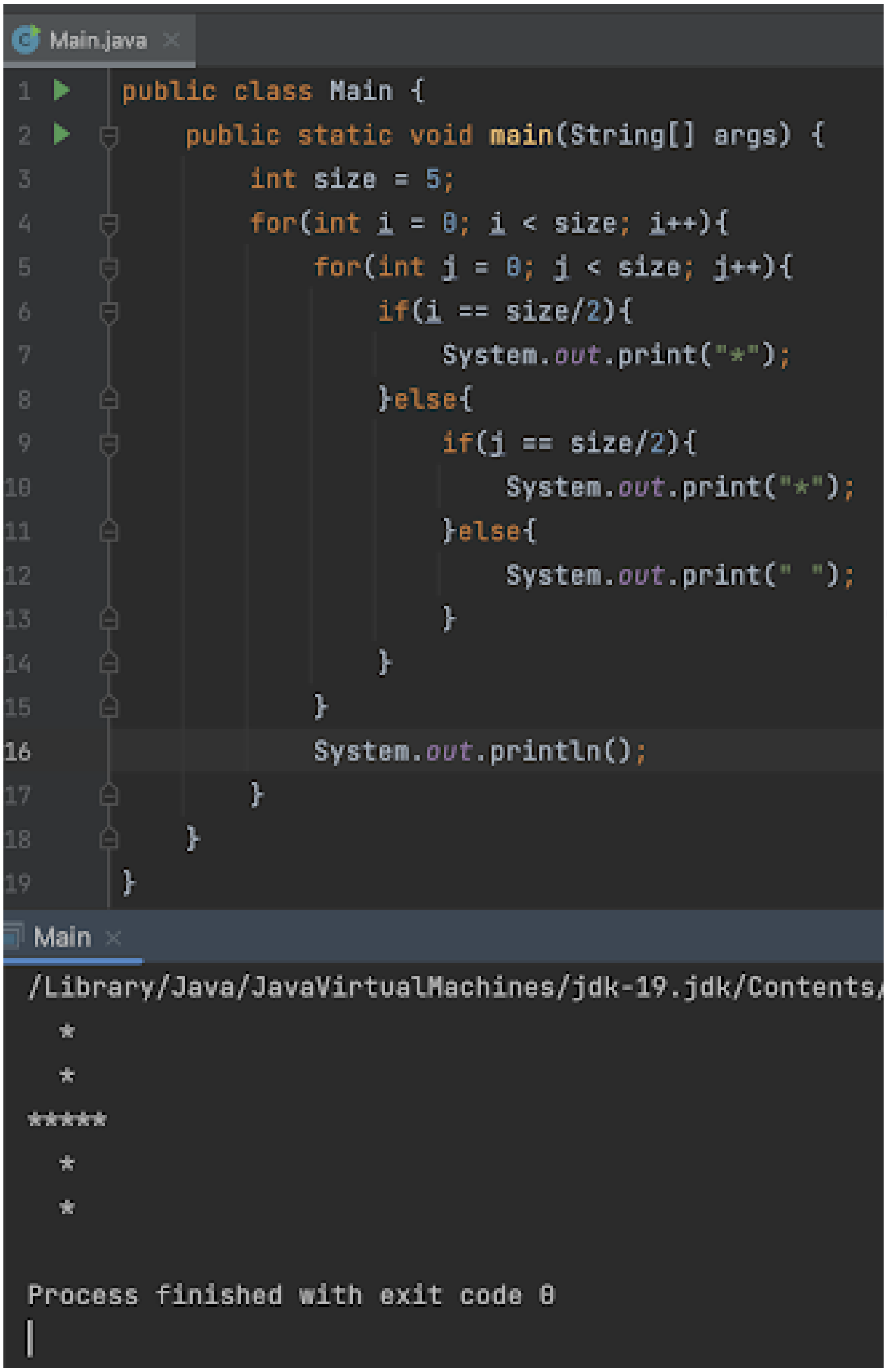
```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java -javaa
    A
   ABC
  ABCDE
 ABCDEFG
ABCDEFGHI
 ABCDEFG
  ABCDE
   ABC
    A

Process finished with exit code 0
```

Q7- Write a program to print pattern given below :

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

Code:

```
public class Test {
   public static void main(String[] args) {
       int size = 5;
       for(int i = 0; i < size; i++){ //representing columns
           for(int j = 0; j < size; j++){//representing rows
               if(i == size/2){ //middle column
                   System.out.print("*");
               }else{
                   if(j == size/2){ //middle row
                       System.out.print("*");
                   }else{
                       System.out.print(" ");
                   }
               }
           }
           System.out.println();
       }
   }
}
```



```java
public class Main {
    public static void main(String[] args) {
        int size = 5;
        for(int i = 0; i < size; i++){
            for(int j = 0; j < size; j++){
                if(i == size/2){
                    System.out.print("*");
                }else{
                    if(j == size/2){
                        System.out.print("*");
                    }else{
                        System.out.print(" ");
                    }
                }
            }
            System.out.println();
        }
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/
  *
  *
*****
  *
  *

Process finished with exit code 0
```

Q8. Write a program to print a triangle of prime numbers upto given number of lines of the triangle.

Input:

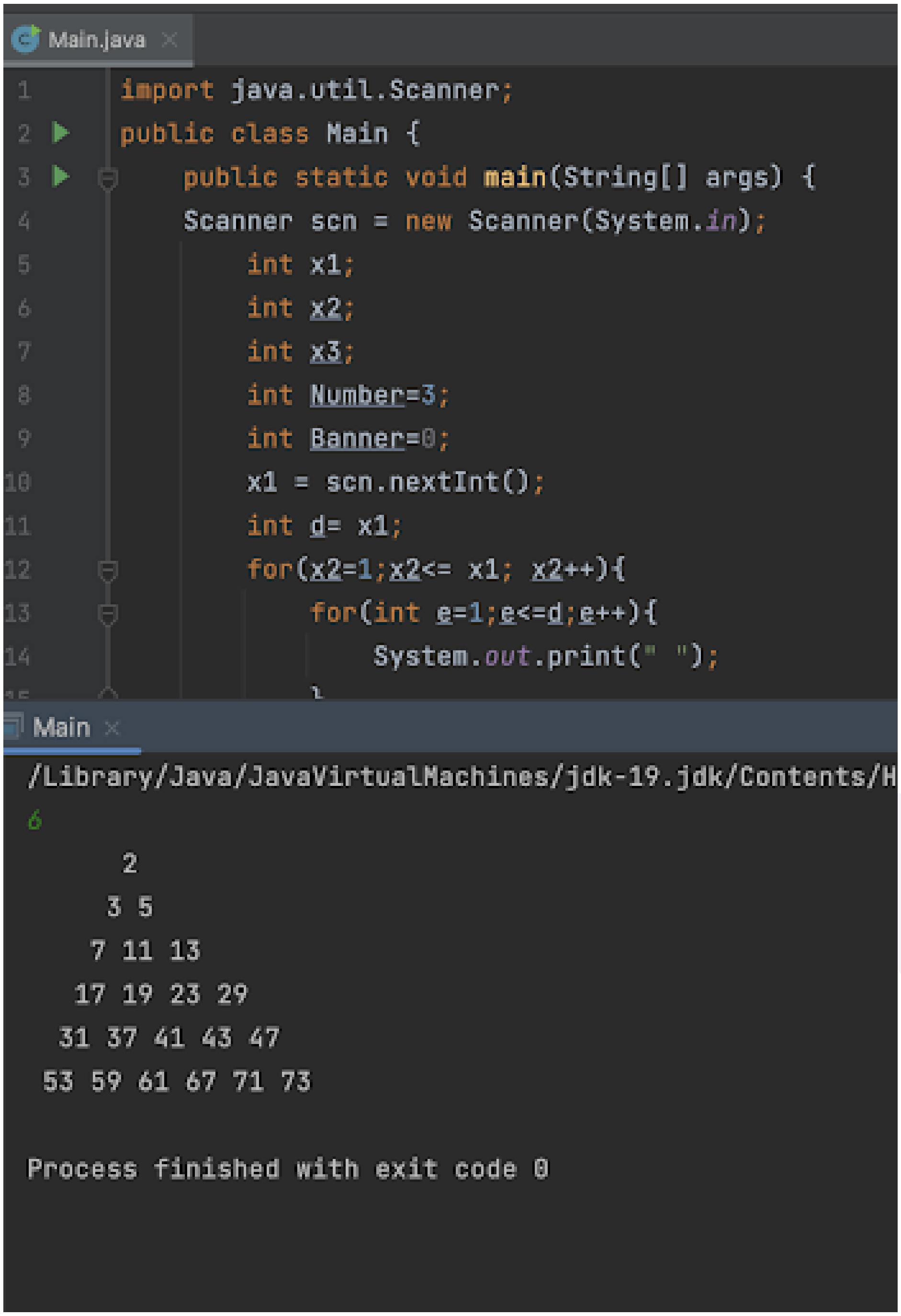
6

Expected Output:

```
      2
     3 5
    7 11 13
  17 19 23 29
 31 37 41 43 47
53 59 61 67 71 73
```

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
   Scanner scn = new Scanner(System.in);
       int x1;
       int x2;
       int x3;
       int number=3;
       int banner=0;
       x1 = scn.nextInt(); //denoting length of pattern
       int d= x1; //number of spaces at each line, equal to line number
       for(x2=1;x2<= x1; x2++){
           for(int e=1;e<=d;e++){
               System.out.print(" ");
           }
           if(x2==1){//1st line
               System.out.print("2");
           }
           else{//other lines
               for(x3=0; x3!= x2;){
                   banner=0;
                   for (int k=2;k<number;k++){ //check if prime or not
                       if((number%k)==0)
                           banner=1; //not prime
                   }
                   if(banner==0){ //prime
                       x3++;
                       System.out.print(number + " ");
                   }
                   number++;
               }
           }
           System.out.println(); //move to next line
           d--;
       }
   }
}
```



```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
        int x1;
        int x2;
        int x3;
        int Number=3;
        int Banner=0;
        x1 = scn.nextInt();
        int d= x1;
        for(x2=1;x2<= x1; x2++){
            for(int e=1;e<=d;e++){
                System.out.print(" ");
            }
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/H
6
      2
     3 5
    7 11 13
   17 19 23 29
  31 37 41 43 47
 53 59 61 67 71 73

Process finished with exit code 0
```

Q9 Write a program to check whether a prime Number can be expressed as a Sum of Two Prime Numbers.

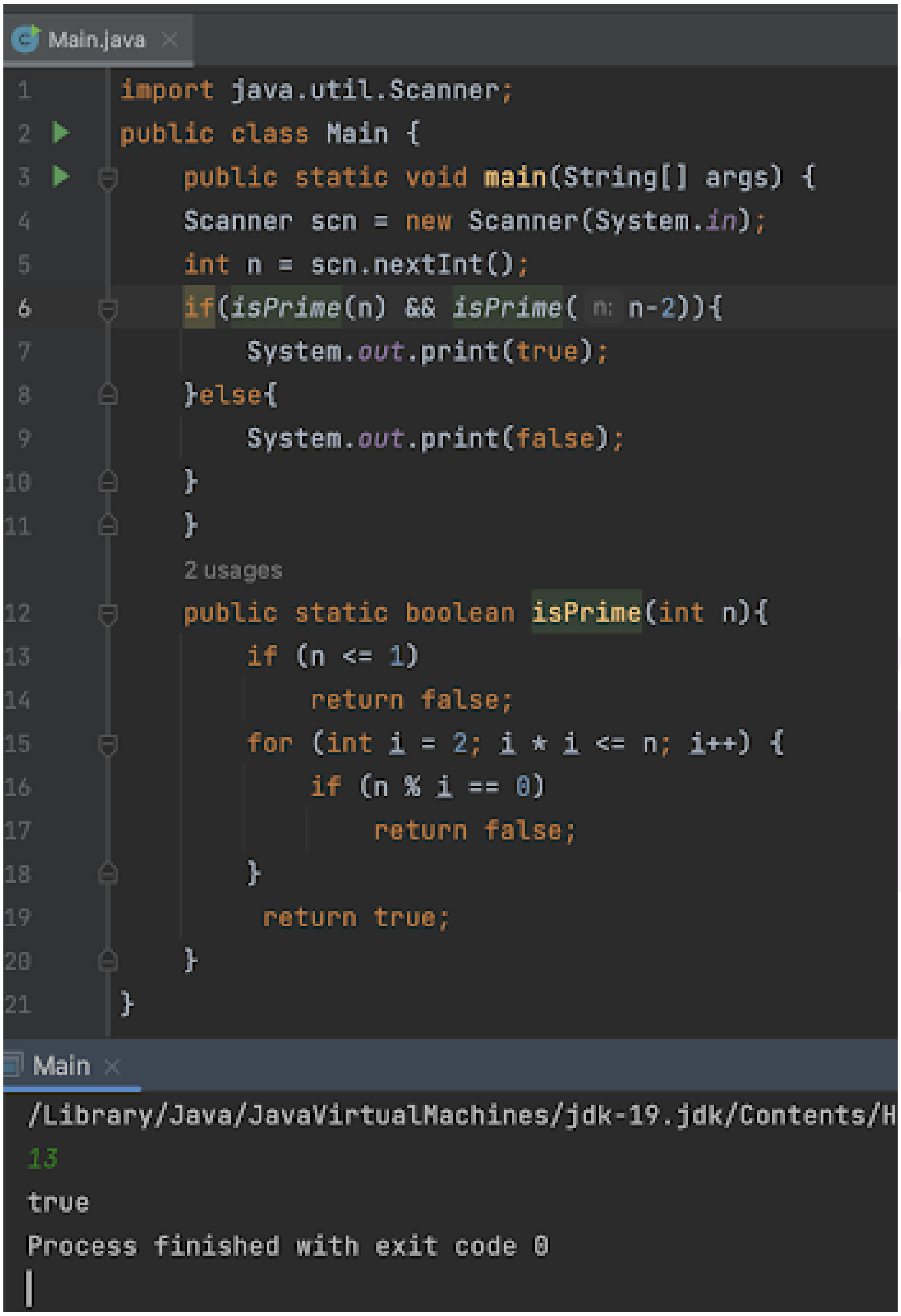
Hint: Apart from 2, all of the prime numbers are odd. So it is not possible to represent a prime number (which is odd) to be written as a sum of two odd prime numbers, so we are sure that one of the two prime numbers should be 2. So we have to check whether n-2 is prime or not.

Input:
13

Expected Output:
True

Code:

```
java.util.Scanner;
public class Test {
   public static void main(String[] args) {
   Scanner scn = new Scanner(System.in);
   int n = scn.nextInt();
   if(isPrimport ime(n) && isPrime(n-2)){
       System.out.print(true);
   }else{
       System.out.print(false);
   }
   }
   public static boolean isPrime(int n){
       if (n <= 1)
           return false;
       for (int i = 2; i * i <= n; i++) {
           if (n % i == 0)//if number is divisible by any number other than 0 and
           itself, denoted by i, it is not a prime number
               return false;
       }
        return true;
   }
}
```



```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    if(isPrime(n) && isPrime( n-2)){
        System.out.print(true);
    }else{
        System.out.print(false);
    }
    }
    public static boolean isPrime(int n){
        if (n <= 1)
            return false;
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0)
                return false;
        }
         return true;
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/H
13
true
Process finished with exit code 0
```

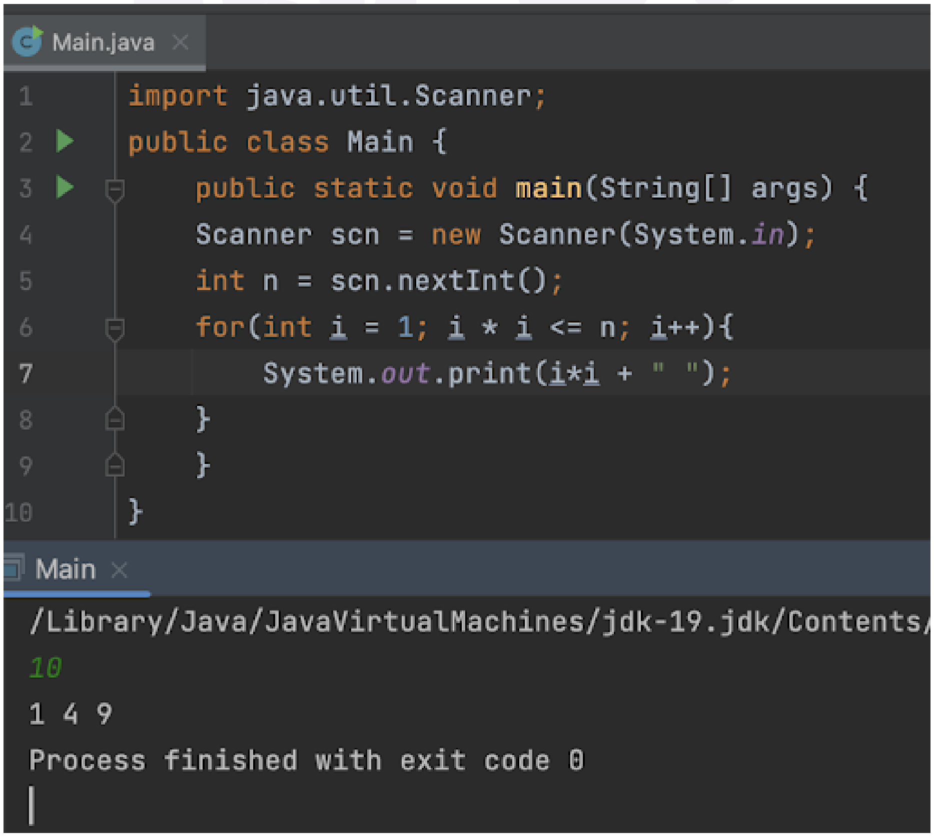
Q10. You are given n number of bulbs. They are all switched off. A weird fluctuation in voltage hits the circuit n times. In the 1st fluctuation all bulbs are toggled, in the 2nd fluctuation every 2nd bulb is toggled, in the 3rd fluctuation every 3rd bulb is toggled and so on. You've to find which bulbs will be switched on after n fluctuations.
Take as input a number n, representing the number of bulbs.
Print all the bulbs that will be on after the nth fluctuation in voltage.

Input:
10

Expected Output:
1 4 9

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
   Scanner scn = new Scanner(System.in);
   int n = scn.nextInt();
   for(int i = 1; i * i <= n; i++){
       System.out.print(i*i + " ");  //only those bulbs will remain on which are perfect
       squares as perfect squares have odd number of divisors due to their square root being
       the extra one, whereas rest have even number of divisors
   }
   }
}
```



```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    for(int i = 1; i * i <= n; i++){
        System.out.print(i*i + " ");
    }
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/
10
1 4 9
Process finished with exit code 0
```