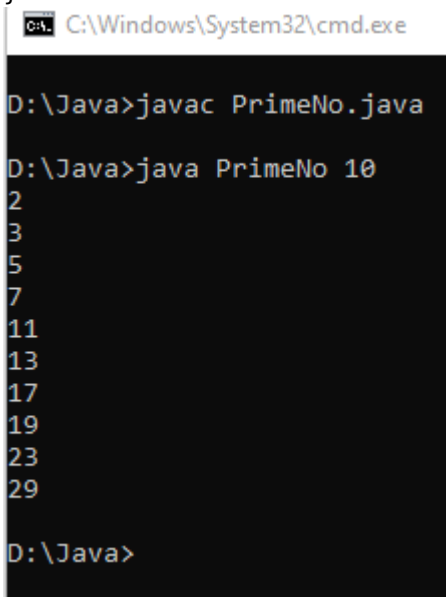


Write Java programs to:

1. Display prime numbers between 1 and 100 or 1 and n

```
class PrimeNo {  
  
    public static void main(String[] args) {  
        int i;  
        int n = Integer.parseInt(args[0]);  
  
        int j = 2;  
        int counter = 0;  
  
        while (counter < n) {  
            int count = 0;  
            for (i = 2; i <= j / 2; i++) {  
                if (j % i == 0) {  
                    count++;  
                }  
            }  
            if (count == 0) {  
                System.out.println(j);  
                counter++;  
            }  
            j++;  
        }  
    }  
}
```



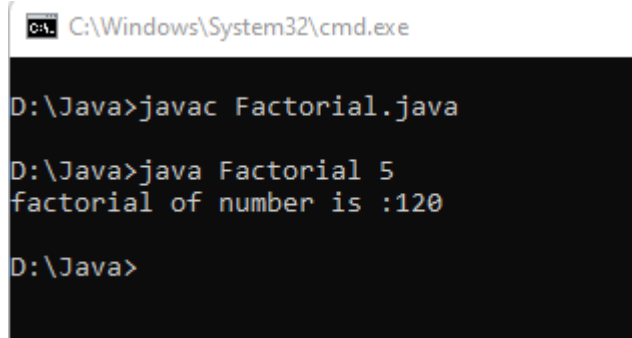
The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe". The user has navigated to the directory "D:\Java" and executed the following commands:

```
D:\Java>javac PrimeNo.java  
  
D:\Java>java PrimeNo 10  
2  
3  
5  
7  
11  
13  
17  
19  
23  
29  
  
D:\Java>
```

The output of the program for n=10 is a list of prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29.

2. Find the factorial of a number

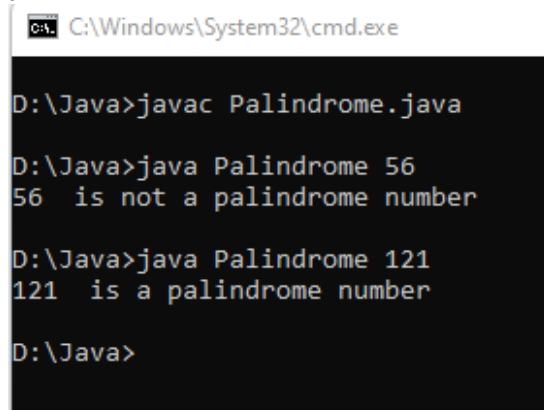
```
class Factorial {  
  
    public static void main(String[] args) {  
        int i;  
        int fact = 1;  
        int n = Integer.parseInt(args[0]);  
        for (i = 1; i <= n; i++) {  
            fact = fact * i;  
        }  
        System.out.println("factorial of number is :" + fact);  
    }  
}
```



```
C:\Windows\System32\cmd.exe  
  
D:\Java>javac Factorial.java  
  
D:\Java>java Factorial 5  
factorial of number is :120  
  
D:\Java>
```

3. Check if a number is palindrome or not

```
class Palindrome {  
  
    public static void main(String[] args) {  
        int rev = 0, rem, pal;  
        int n = Integer.parseInt(args[0]);  
        pal = n;  
        while (n != 0) {  
            rem = n % 10;  
            rev = rev * 10 + rem;  
            n = n / 10;  
        }  
        if (pal == rev)  
            System.out.println(pal + " " + " is a palindrome number");  
        else  
            System.out.println(pal + " " + " is not a palindrome number");  
    }  
}
```



```
C:\Windows\System32\cmd.exe  
  
D:\Java>javac Palindrome.java  
  
D:\Java>java Palindrome 56  
56 is not a palindrome number  
  
D:\Java>java Palindrome 121  
121 is a palindrome number  
  
D:\Java>
```

4. Add two integer variables in 5 different ways using methods and control statement

```
class AddUsingMethods {

    public static void main(String args[]) {
        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);

        Addition add = new Addition(x, y);

        int sum = add.addWithArgsAndReturn(x, y);
        System.out.println("Addition with Args And Return: " + sum);

        add.addWithArgsAndNoReturn(x, y);

        int sum2 = add.addwithNoArgsAndReturn();
        System.out.println("Addition with No Args And Return: " + sum2);

        add.addwithNoArgsNoReturn();
    }
}

class Addition {

    int x;
    int y;

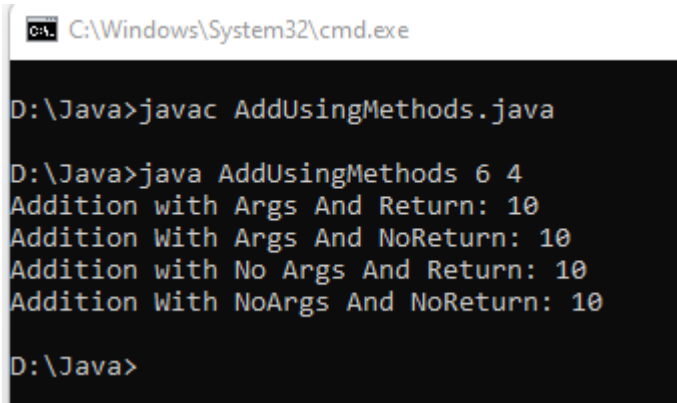
    Addition(int x1, int y1) {
        x = x1;
        y = y1;
    }

    public int addWithArgsAndReturn(int a, int b) {
        return a + b;
    }

    public void addWithArgsAndNoReturn(int a, int b) {
        System.out.println("Addition With Args And NoReturn: " + (a + b));
    }

    public int addwithNoArgsAndReturn() {
        return x + y;
    }

    public void addwithNoArgsNoReturn() {
        System.out.println("Addition With NoArgs And NoReturn: " + (x + y));
    }
}
```



```
C:\Windows\System32\cmd.exe

D:\Java>javac AddUsingMethods.java

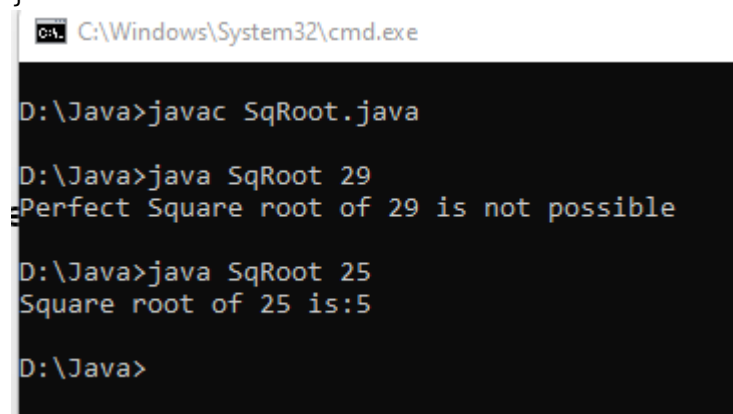
D:\Java>java AddUsingMethods 6 4
Addition with Args And Return: 10
Addition With Args And NoReturn: 10
Addition with No Args And Return: 10
Addition With NoArgs And NoReturn: 10

D:\Java>
```

5. Find square root of a number without sqrt method

```
class SqRoot {

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int i;
        boolean flag = false;
        for (i = 1; i <= n; i++) {
            int sqRoot = n / i;
            if (sqRoot == i && n % i == 0) {
                System.out.println("Square root of " + n + " is:" + sqRoot);
                flag = true;
                break;
            }
        }
        if (!flag) {
            System.out.println("Perfect Square root of " + n + " is not
possible");
        }
    }
}
```



```
C:\Windows\System32\cmd.exe

D:\Java>javac SqRoot.java

D:\Java>java SqRoot 29
Perfect Square root of 29 is not possible

D:\Java>java SqRoot 25
Square root of 25 is:5

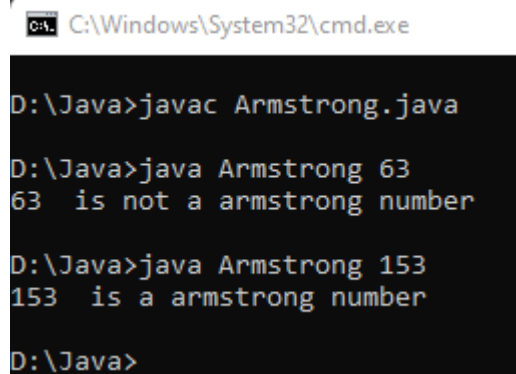
D:\Java>
```

6. Program to Check Armstrong number

```
class Armstrong {

    public static void main(String[] args) {
        int sum = 0, r, armstrong;
        int n = Integer.parseInt(args[0]);
        armstrong = n;
```

```
while (n != 0) {
    r = n % 10;
    sum = sum + (r * r * r);
    n = n / 10;
}
if (armstrong == sum) {
    System.out.println(armstrong + " " + " is a armstrong number");
} else {
    System.out.println(armstrong + " " + " is not a armstrong number");
}
}
```



```
C:\Windows\System32\cmd.exe

D:\Java>javac Armstrong.java

D:\Java>java Armstrong 63
63 is not a armstrong number

D:\Java>java Armstrong 153
153 is a armstrong number

D:\Java>
```

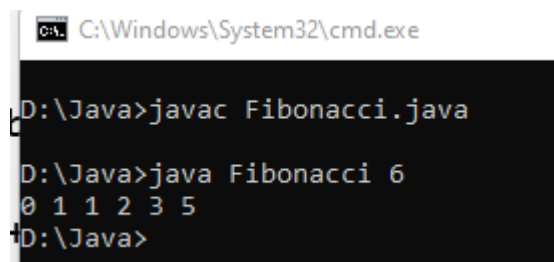
7. Print Fibonacci series till n

```
class Fibonacci {

    public static void main(String[] args) {
        int n1 = 0, n2 = 1, n3, i;
        System.out.print(n1 + " " + n2); //printing 0 and 1

        int n = Integer.parseInt(args[0]);

        for (i = 2; i < n; ++i) {
            //loop starts from 2 because 0 and 1 are already printed
            n3 = n1 + n2;
            System.out.print(" " + n3);
            n1 = n2;
            n2 = n3;
        }
    }
}
```



```
C:\Windows\System32\cmd.exe

D:\Java>javac Fibonacci.java

D:\Java>java Fibonacci 6
0 1 1 2 3 5

D:\Java>
```

8. Program to find sum of digits of a number

```
class SumOfDigits {  
  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        int sum = 0, r;  
        while (n != 0) {  
            r = n % 10;  
            sum = sum + r;  
            n = n / 10;  
        }  
        System.out.println("Sum of digits :" + sum);  
    }  
}
```

C:\windows\system32\cmd.exe

D:\Java>javac SumOfDigits.java

D:\Java>java SumOfDigits 675
Sum of digits :18

D:\Java>java SumOfDigits 6758
Sum of digits :26

D:\Java>

9. Program to print math tables in a given range (11 - 25 tables)

```
class MultiplicationTable {  
  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        int m = Integer.parseInt(args[1]);  
        int mult;  
        for (int j = n; j <= m; j++) {  
            for (int i = 1; i <= 10; i++) {  
                mult = j * i;  
                System.out.print(j + "*" + i + "=" + mult + " ");  
            }  
            System.out.println();  
            System.out.println();  
        }  
    }  
}
```

```

D:\Java>javac MultiplicationTable.java

D:\Java>java MultiplicationTable 11 25
11*1=11 11*2=22 11*3=33 11*4=44 11*5=55 11*6=66 11*7=77 11*8=88 11*9=99 11*10=110
12*1=12 12*2=24 12*3=36 12*4=48 12*5=60 12*6=72 12*7=84 12*8=96 12*9=108 12*10=120
13*1=13 13*2=26 13*3=39 13*4=52 13*5=65 13*6=78 13*7=91 13*8=104 13*9=117 13*10=130
14*1=14 14*2=28 14*3=42 14*4=56 14*5=70 14*6=84 14*7=98 14*8=112 14*9=126 14*10=140
15*1=15 15*2=30 15*3=45 15*4=60 15*5=75 15*6=90 15*7=105 15*8=120 15*9=135 15*10=150
16*1=16 16*2=32 16*3=48 16*4=64 16*5=80 16*6=96 16*7=112 16*8=128 16*9=144 16*10=160
17*1=17 17*2=34 17*3=51 17*4=68 17*5=85 17*6=102 17*7=119 17*8=136 17*9=153 17*10=170
18*1=18 18*2=36 18*3=54 18*4=72 18*5=90 18*6=108 18*7=126 18*8=144 18*9=162 18*10=180
19*1=19 19*2=38 19*3=57 19*4=76 19*5=95 19*6=114 19*7=133 19*8=152 19*9=171 19*10=190
20*1=20 20*2=40 20*3=60 20*4=80 20*5=100 20*6=120 20*7=140 20*8=160 20*9=180 20*10=200
21*1=21 21*2=42 21*3=63 21*4=84 21*5=105 21*6=126 21*7=147 21*8=168 21*9=189 21*10=210
22*1=22 22*2=44 22*3=66 22*4=88 22*5=110 22*6=132 22*7=154 22*8=176 22*9=198 22*10=220
23*1=23 23*2=46 23*3=69 23*4=92 23*5=115 23*6=138 23*7=161 23*8=184 23*9=207 23*10=230
24*1=24 24*2=48 24*3=72 24*4=96 24*5=120 24*6=144 24*7=168 24*8=192 24*9=216 24*10=240
25*1=25 25*2=50 25*3=75 25*4=100 25*5=125 25*6=150 25*7=175 25*8=200 25*9=225 25*10=250

D:\Java>

```

10. Write a program to give the examples of operators.

- 1) Increment and decrement operators.
- 2) Arithmetic operator.
- 3) Relational Operator
- 4) Bitwise operator.
- 5) Conditional Operator

```

public class Operator {
    public static void main(String[] args) {
        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);

        System.out.println("1. Increment / Decrement Operator");
        System.out.println("PreIncrement: ++" + x + " = " + (++x));
        System.out.println("Post Increment: " + x + "++ = " + (x++));
        System.out.println("Pre Decrement: --" + y + " = " + (--y));
        System.out.println("Post Decrement: " + y + "-- = " + (y--));
        System.out.println();

        System.out.println("2. Arithmetic Operator");
        System.out.println("Addition: " + x + " + " + y + " = " + (x + y));
        System.out.println("Substraction: " + x + " - " + y + " = " + (x -
y));
        System.out.println("Multiplication: " + x + " * " + y + " = " + (x
* y));
        System.out.println("Division: " + x + " / " + y + " = " + (x / y));
        System.out.println("Modulus: " + x + " % " + y + " = " + (x % y));
        System.out.println();
    }
}

```

```
        System.out.println("3. Relational Operator");
        System.out.println("Equals: " + x + " == " + y + " = " + (x == y));
        System.out.println("Not Equal: " + x + " != " + y + " = " + (x !=
y));
        System.out.println("Greater Than: " + x + " > " + y + " = " + (x >
y));
        System.out.println("Less than: " + x + " < " + y + " = " + (x <
y));
        System.out.println("Greater than Equal to: " + x + " >= " + y + " =
" + (x >= y));
        System.out.println("Less than equal to: " + x + " <= " + y + " = "
+ (x <= y));
        System.out.println();

        System.out.println("4. Bitwise Operator");
        System.out.println("AND: " + x + " & " + y + " = " + (x & y));
        System.out.println("OR: " + x + " | " + y + " = " + (x | y));
        System.out.println("XOR: " + x + " ^ " + y + " = " + (x ^ y));
        System.out.println("NOT: ~" + y + " = " + (~y));
        System.out.println("LEFT SHIFT: " + x + " << " + 1 + " = " + (x <<
1));
        System.out.println("RIGHT SHIFT: " + x + " >> " + 2 + " = " + (x >>
1));
        System.out.println("zero-fill right shift: " + x + " >>> " + 1 + "
= " + (x >>> 1));
        System.out.println();

        System.out.println("4. Conditional Operator");
        System.out.println("Logical AND: " + x + " > 1 && " + y + " < 1
returns " + (x > 1 && y < 1));
        System.out.println("Logical OR: " + x + " > 1 || " + y + " < 1
returns " + (x > 1 || y < 1));
        System.out.println("Ternary: (" + x + " > 5 ? GreaterThan5 :
LessThanEqualTo5) returns "
+ (x > 5 ? "GreaterThan5" : "LessThanEqualTo5"));
        System.out.println();

    }

}
```



```
D:\Java>javac Operator.java

D:\Java>java Operator 7 4
1. Increment / Decrement Operator
PreIncrement: ++7 = 8
Post Increment: 8++ = 8
Pre Decrement: --4 = 3
Post Decrement: 3-- = 3

2. Arithmetic Operator
Addition: 9 + 2 = 11
Subtraction: 9 - 2 = 7
Multiplication: 9 * 2 = 18
Division: 9 / 2 = 4
Modulus: 9 % 2 = 1

3. Relational Operator
Equals: 9 == 2 = false
Not Equal: 9 != 2 = true
Greater Than: 9 > 2 = true
Less than: 9 < 2 = false
Greater than Equal to: 9 >= 2 = true
Less than equal to: 9 <= 2 = false

4. Bitwise Operator
AND: 9 & 2 = 0
OR: 9 | 2 = 11
XOR: 9 ^ 2 = 11
NOT: ~2 = -3
LEFT SHIFT: 9 << 1 = 18
RIGHT SHIFT: 9 >> 2 = 4
zero-fill right shift: 9 >>> 1 = 4

4. Conditional Operator
Logical AND: 9 > 1 && 2 < 1 returns false
Logical OR: 9 > 1 || 2 < 1 returns true
Ternary: (9 > 5 ? GreaterThan5 : LessThanEqualTo5) returns GreaterThan5

D:\Java>
```