Name : Mohamed Aashir S

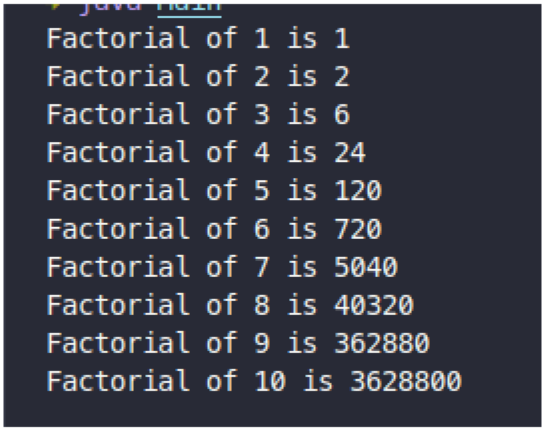
Roll no : 22pt17

Subject : JAVA Programming

Problem_Sheet _2

1.Write an application that displays the factorial for every integer value from 1 to 10. A factorial of a number is the product of that number multiplied by each positive integer lower than it.

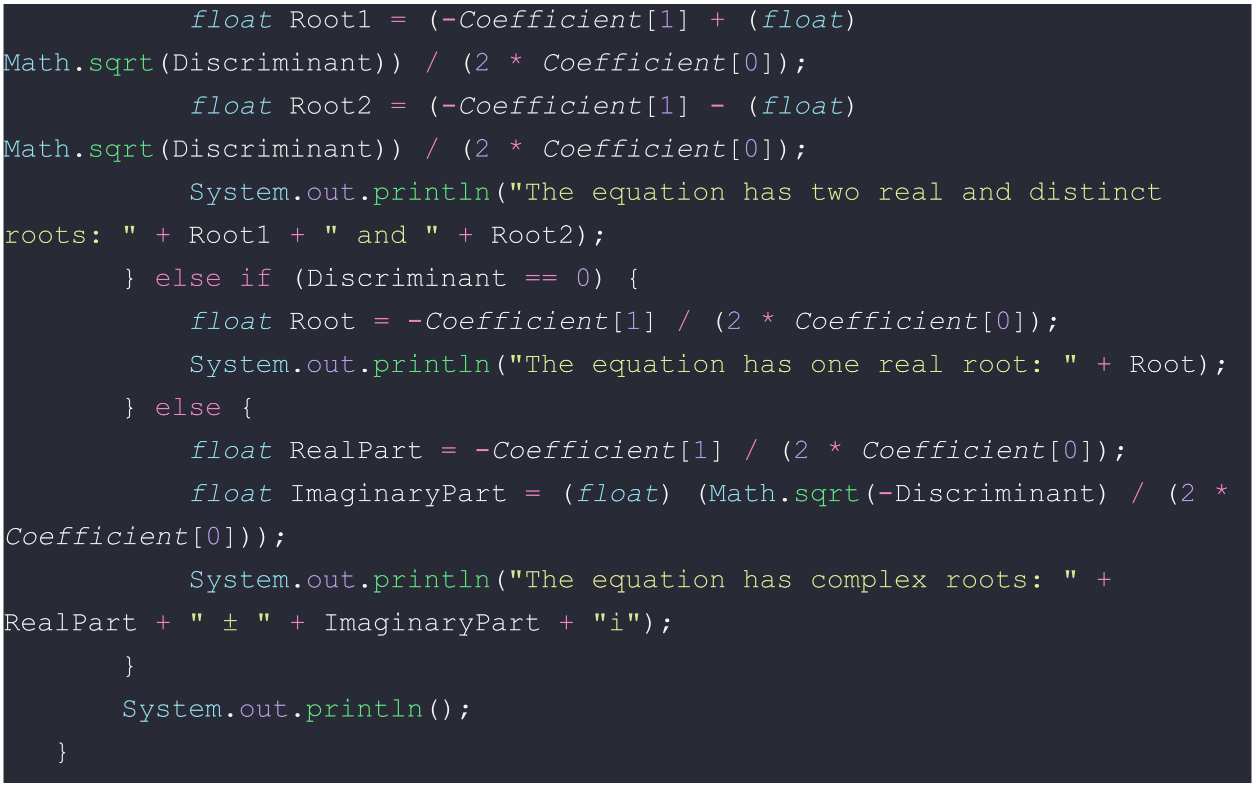
```
   private static void Factorial() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        Function = n -> n == 0 ? 1 : n * Function.applyAsInt(n - 1);
        for (int i = 1; i <= 10; i++)
            System.out.println("Factorial of " + i + " is " +
Function.applyAsInt(i));
        System.out.println();
   }
```



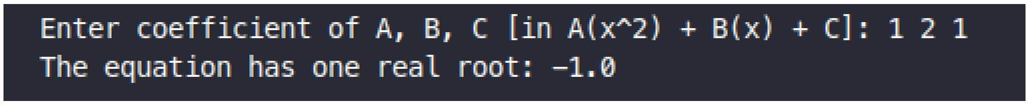
```
java Main
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
Factorial of 8 is 40320
Factorial of 9 is 362880
Factorial of 10 is 3628800
```

2. Write a Java program to solve quadratic equations (use if, else if and else).

```
   private static void Solve() {
        System.out.print("Enter coefficient of A, B, C [in A(x^2) + B(x) +
C]: ");
        float[] Coefficient = { input.nextFloat(), input.nextFloat(),
input.nextFloat() };
        float Discriminant = (Coefficient[1] * Coefficient[1]) - (4 *
Coefficient[0] * Coefficient[2]);
        if (Discriminant > 0) {
```

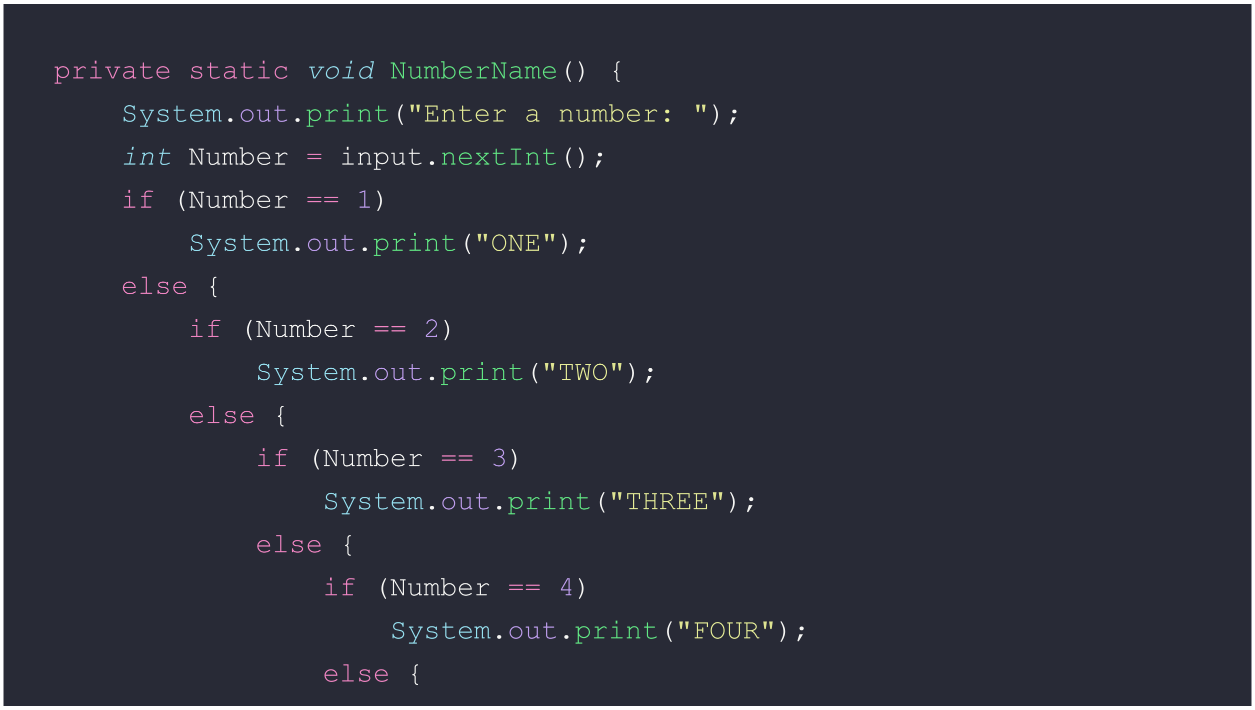


```
           float Root1 = (-Coefficient[1] + (float)
Math.sqrt(Discriminant)) / (2 * Coefficient[0]);
           float Root2 = (-Coefficient[1] - (float)
Math.sqrt(Discriminant)) / (2 * Coefficient[0]);
           System.out.println("The equation has two real and distinct
roots: " + Root1 + " and " + Root2);
       } else if (Discriminant == 0) {
           float Root = -Coefficient[1] / (2 * Coefficient[0]);
           System.out.println("The equation has one real root: " + Root);
       } else {
           float RealPart = -Coefficient[1] / (2 * Coefficient[0]);
           float ImaginaryPart = (float) (Math.sqrt(-Discriminant) / (2 *
Coefficient[0]));
           System.out.println("The equation has complex roots: " +
RealPart + " ± " + ImaginaryPart + "i");
       }
       System.out.println();
   }
```

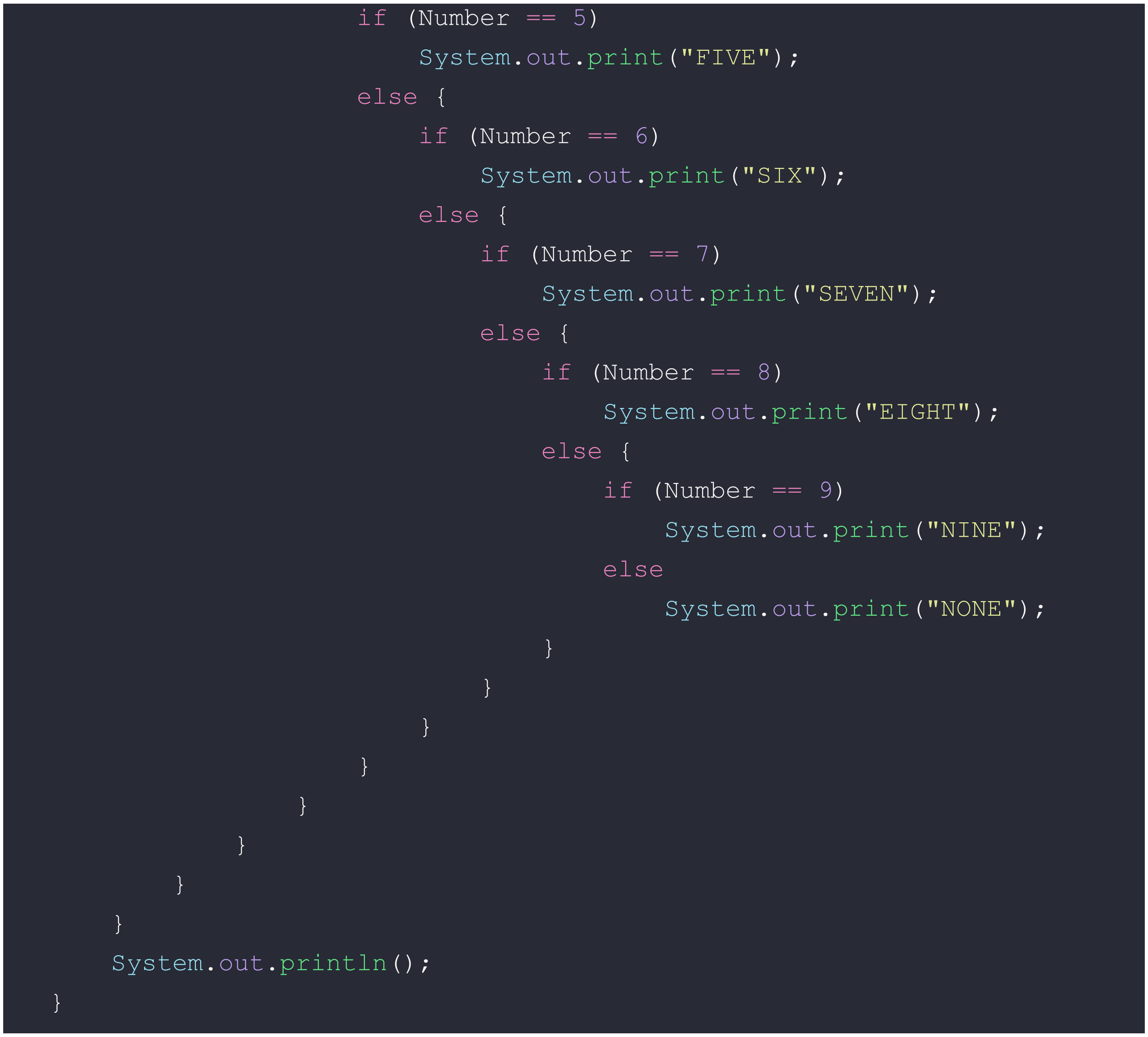


```
Enter coefficient of A, B, C [in A(x^2) + B(x) + C]: 1 2 1
The equation has one real root: -1.0
```

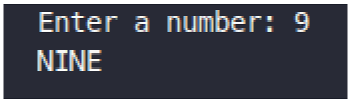
3. Write a program which prints "ONE", "TWO",... , "NINE", "OTHER" if the int variable "number" is 1, 2,... , 9, or other, respectively. Use "nested-if" statement.



```
   private static void NumberName() {
       System.out.print("Enter a number: ");
       int Number = input.nextInt();
       if (Number == 1)
           System.out.print("ONE");
       else {
           if (Number == 2)
               System.out.print("TWO");
           else {
               if (Number == 3)
                   System.out.print("THREE");
               else {
                   if (Number == 4)
                       System.out.print("FOUR");
                   else {
```

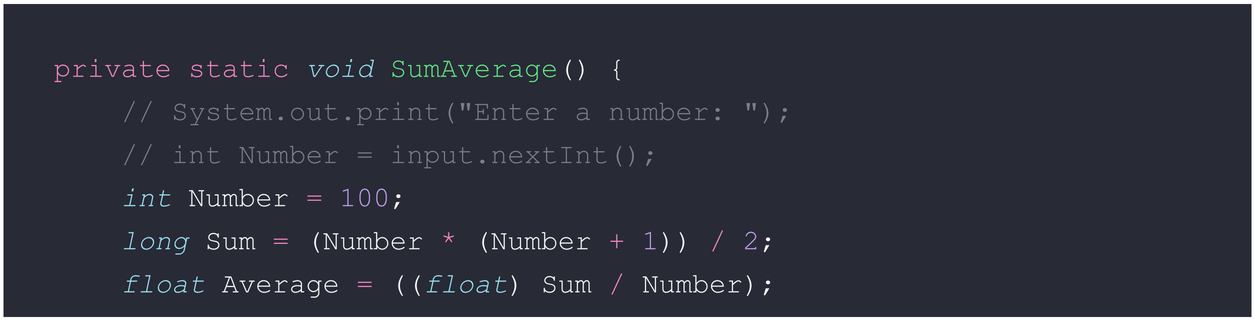


```
                    if (Number == 5)
                        System.out.print("FIVE");
                    else {
                        if (Number == 6)
                            System.out.print("SIX");
                        else {
                            if (Number == 7)
                                System.out.print("SEVEN");
                            else {
                                if (Number == 8)
                                    System.out.print("EIGHT");
                                else {
                                    if (Number == 9)
                                        System.out.print("NINE");
                                    else
                                        System.out.print("NONE");
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    System.out.println();
}
```

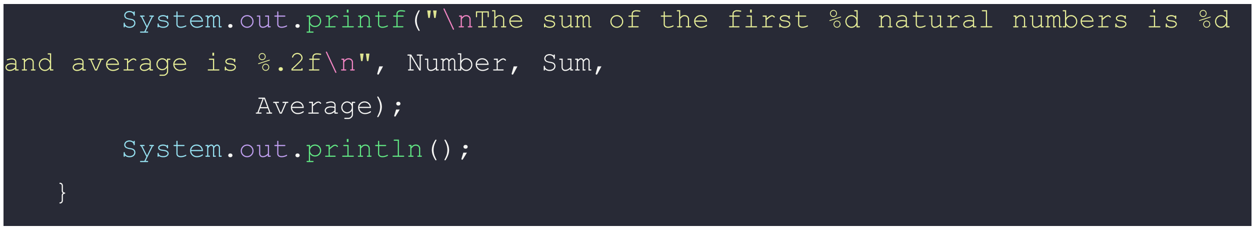


```
Enter a number: 9
NINE
```

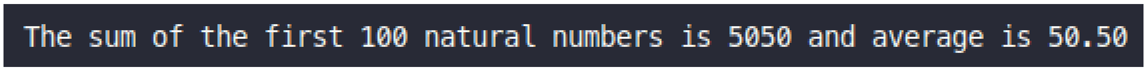
4. Write a program called to produce the sum of 1, 2, 3, ..., to 100. Also compute and display the average.



```
private static void SumAverage() {
    // System.out.print("Enter a number: ");
    // int Number = input.nextInt();
    int Number = 100;
    long Sum = (Number * (Number + 1)) / 2;
    float Average = ((float) Sum / Number);
```

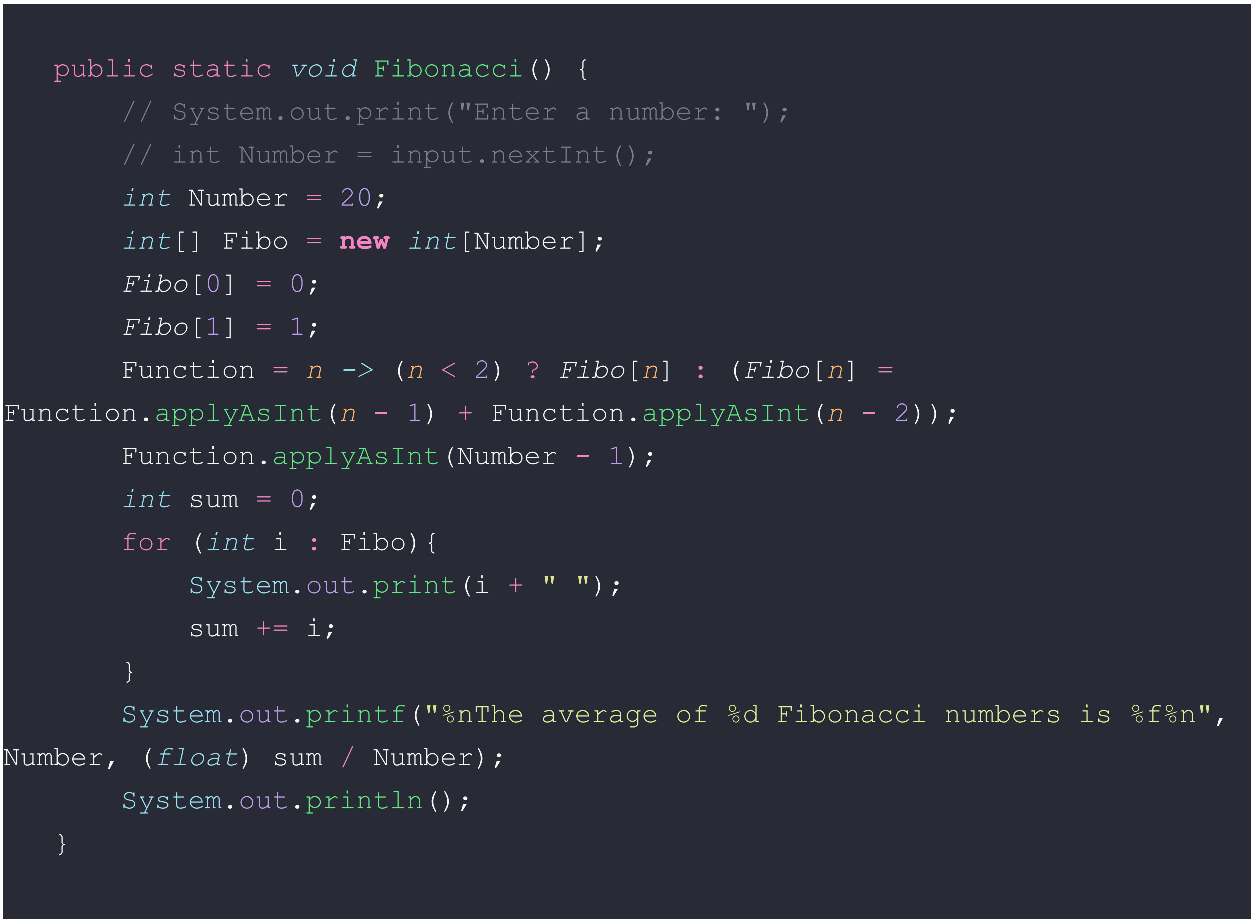


```
        System.out.printf("\nThe sum of the first %d natural numbers is %d
and average is %.2f\n", Number, Sum,
                Average);
        System.out.println();
    }
```

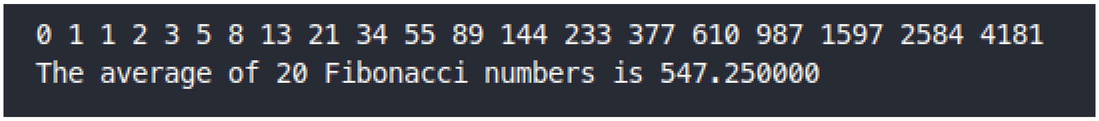


```
The sum of the first 100 natural numbers is 5050 and average is 50.50
```

5. Write a program to print the first 20 Fibonacci numbers F(n), where F(n)=F(n–1)+F(n–2) and F(1)=F(2)=1. Also compute their average.

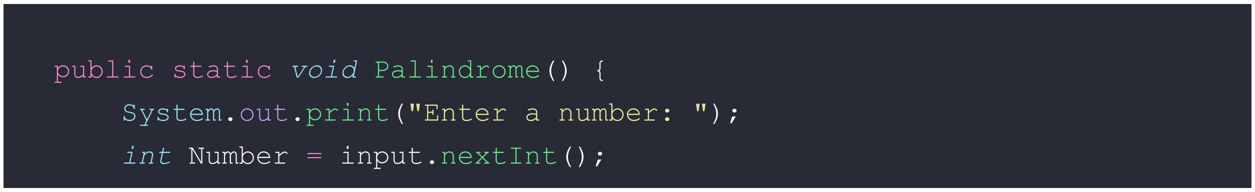


```
    public static void Fibonacci() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        int Number = 20;
        int[] Fibo = new int[Number];
        Fibo[0] = 0;
        Fibo[1] = 1;
        Function = n -> (n < 2) ? Fibo[n] : (Fibo[n] =
Function.applyAsInt(n - 1) + Function.applyAsInt(n - 2));
        Function.applyAsInt(Number - 1);
        int sum = 0;
        for (int i : Fibo){
            System.out.print(i + " ");
            sum += i;
        }
        System.out.printf("%nThe average of %d Fibonacci numbers is %f%n",
Number, (float) sum / Number);
        System.out.println();
    }
```

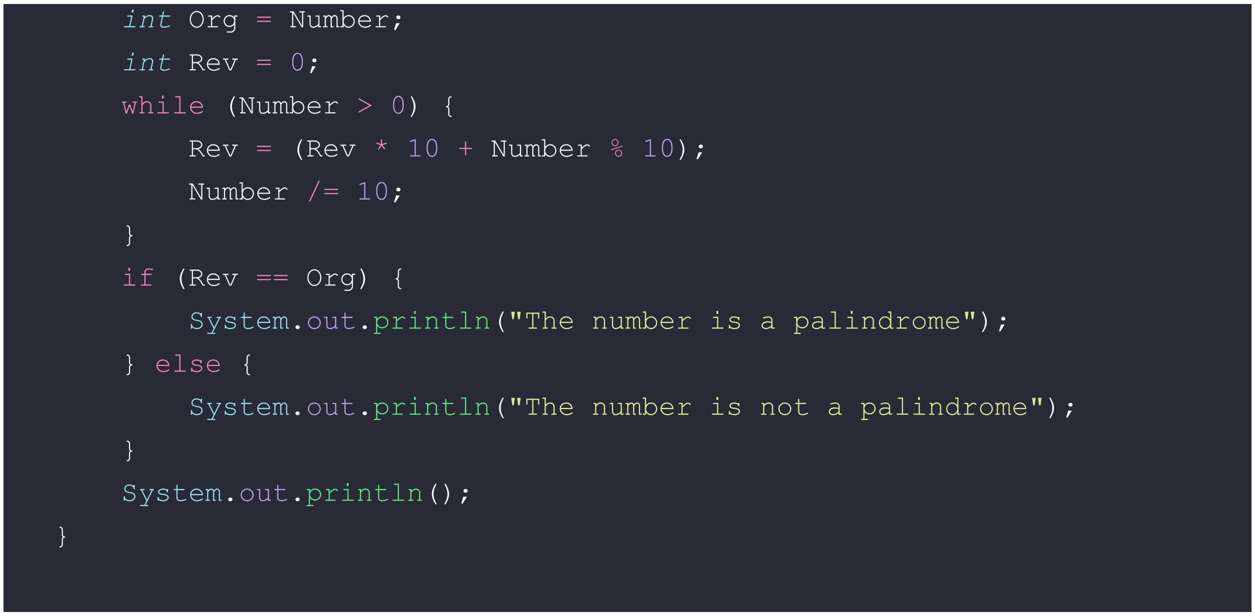


```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
The average of 20 Fibonacci numbers is 547.250000
```

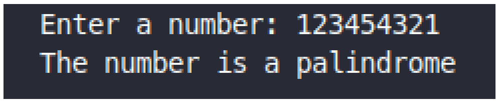
6. Write a program to compute whether the given integer number is a palindrome.



```
    public static void Palindrome() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
```

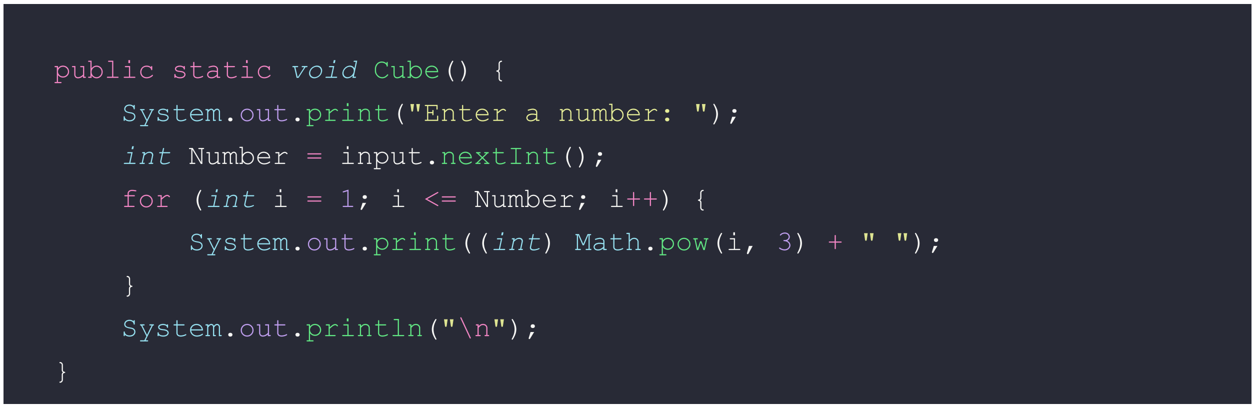


```
        int Org = Number;
        int Rev = 0;
        while (Number > 0) {
            Rev = (Rev * 10 + Number % 10);
            Number /= 10;
        }
        if (Rev == Org) {
            System.out.println("The number is a palindrome");
        } else {
            System.out.println("The number is not a palindrome");
        }
        System.out.println();
    }
```

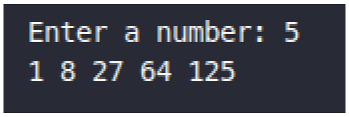


```
Enter a number: 123454321
The number is a palindrome
```

7. Write a program in Java to display the cube of the number up to given an integer.

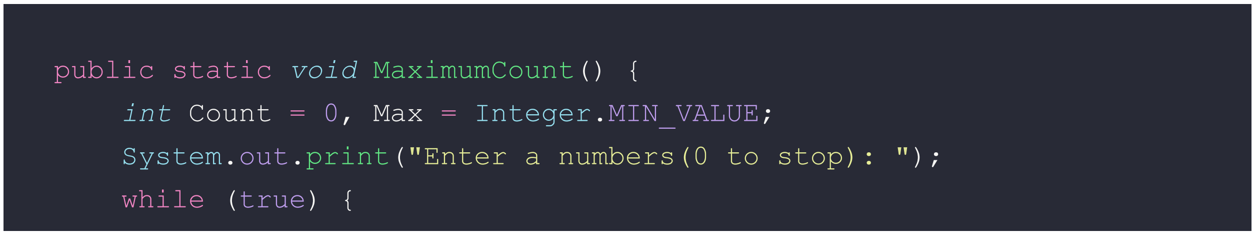


```
    public static void Cube() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
        for (int i = 1; i <= Number; i++) {
            System.out.print((int) Math.pow(i, 3) + " ");
        }
        System.out.println("\n");
    }
```

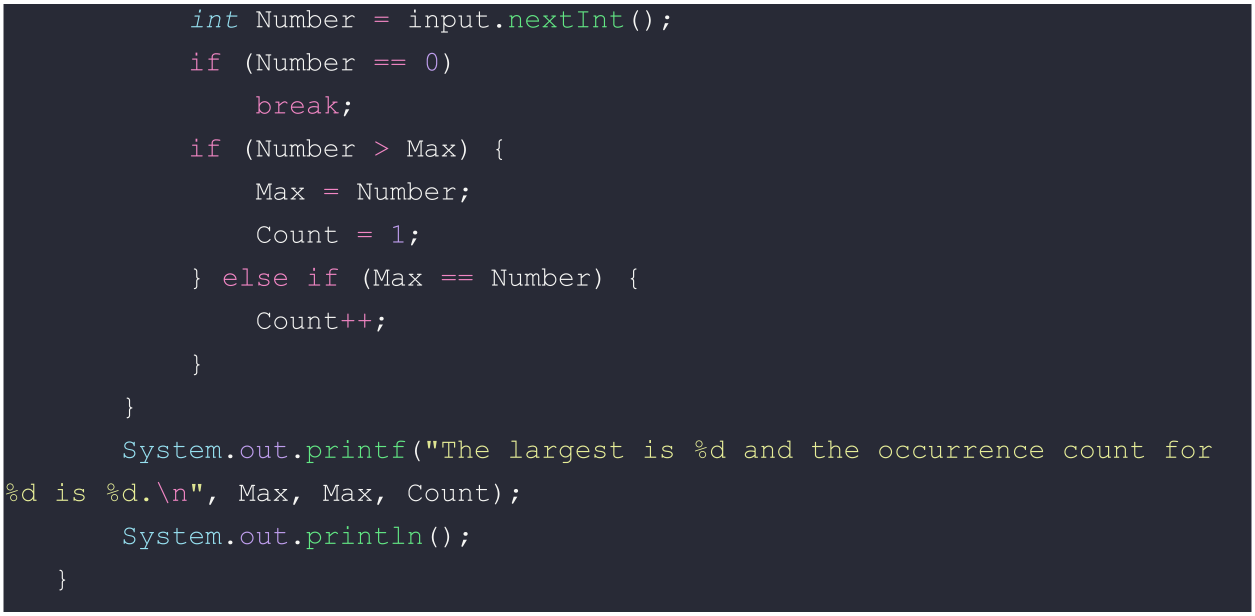


```
Enter a number: 5
1 8 27 64 125
```

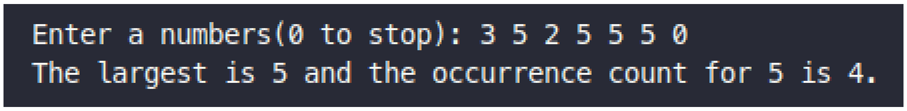
8. Write a program that reads integers, finds the largest of them, and counts its occurrences. Assume that the input ends with number 0. Suppose that you entered 3 5 2 5 5 5 0; the program finds that the largest is 5 and the occurrence count for 5 is 4.



```
    public static void MaximumCount() {
        int Count = 0, Max = Integer.MIN_VALUE;
        System.out.print("Enter a numbers(0 to stop): ");
        while (true) {
```



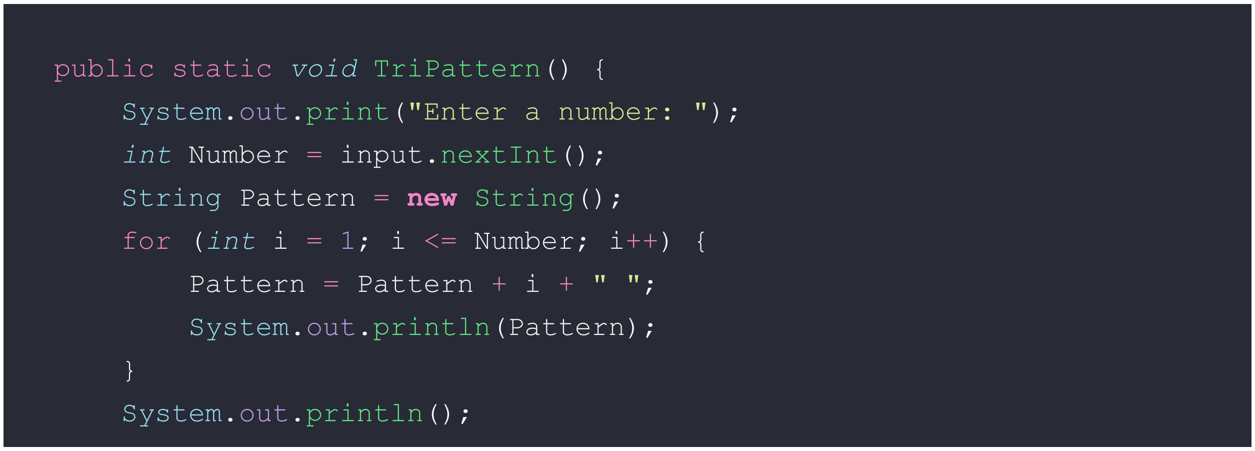
```
        int Number = input.nextInt();
        if (Number == 0)
            break;
        if (Number > Max) {
            Max = Number;
            Count = 1;
        } else if (Max == Number) {
            Count++;
        }
    }
    System.out.printf("The largest is %d and the occurrence count for
%d is %d.\n", Max, Max, Count);
    System.out.println();
}
```



```
Enter a numbers(0 to stop): 3 5 2 5 5 5 0
The largest is 5 and the occurrence count for 5 is 4.
```

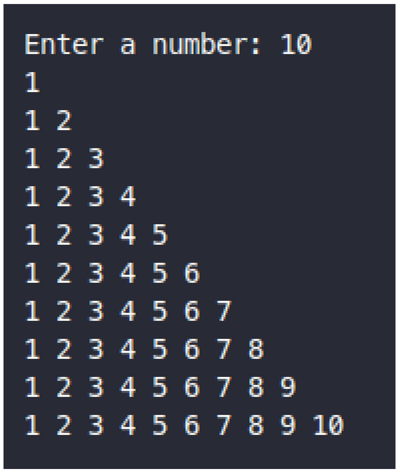
9. Write a program in Java to display the pattern like right angle triangle with a number.

1
12
123
1234
12345
123456
1234567
12345678
123456789
12345678910



```
public static void TriPattern() {
    System.out.print("Enter a number: ");
    int Number = input.nextInt();
    String Pattern = new String();
    for (int i = 1; i <= Number; i++) {
        Pattern = Pattern + i + " ";
        System.out.println(Pattern);
    }
    System.out.println();
```

```
  }
```

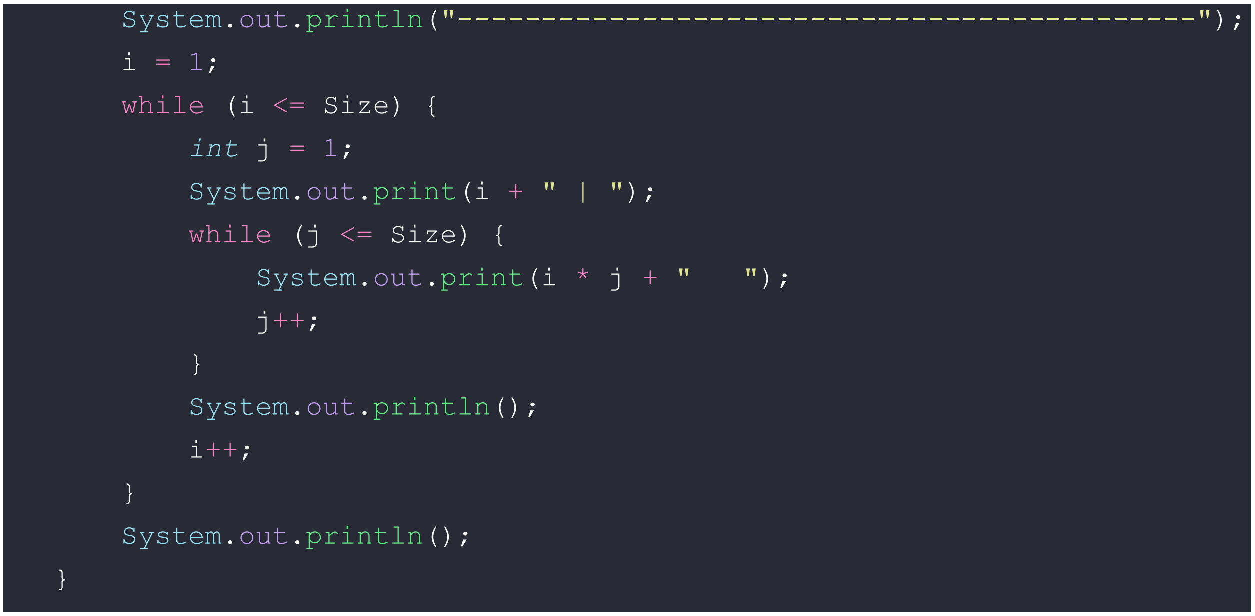


```
Enter a number: 10
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

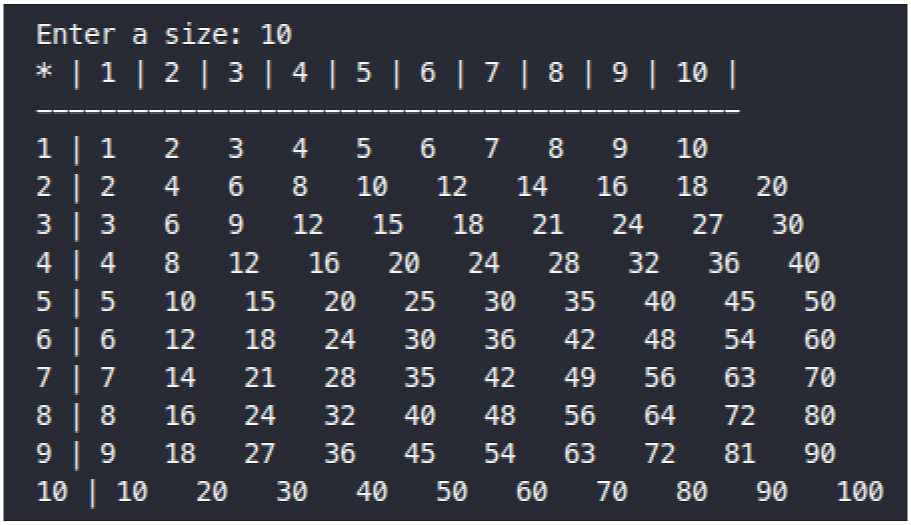
10. Write a program that prompts user for the size (a positive integer in int); and prints the multiplication table as shown: (Use Sentinel-controlled loop)

```
Enter the size: 10
* | 1 2 3 4 5 6 7 8 9 10
1 | 1 2 3 4 5 6 7 8 9 10
2 | 2 4 6 8 10 12 14 16 18 20
3 | 3 6 9 12 15 18 21 24 27 30
4 | 4 8 12 16 20 24 28 32 36 40
5 | 5 10 15 20 25 30 35 40 45 50
6 | 6 12 18 24 30 36 42 48 54 60
7 | 7 14 21 28 35 42 49 56 63 70
8 | 8 16 24 32 40 48 56 64 72 80
9 | 9 18 27 36 45 54 63 72 81 90
10 | 10 20 30 40 50 60 70 80 90 100
```

```java
public static void Table() {
    System.out.print("Enter a size: ");
    int Size = input.nextInt();
    int i = 1;
    System.out.print("* | ");
    while (i <= Size) {
        System.out.print(i + " | ");
        i++;
    }
    System.out.println();
```



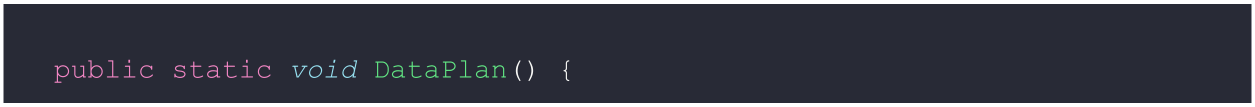
```java
        System.out.println("--------------------------------------------");
        i = 1;
        while (i <= Size) {
            int j = 1;
            System.out.print(i + " | ");
            while (j <= Size) {
                System.out.print(i * j + "   ");
                j++;
            }
            System.out.println();
            i++;
        }
        System.out.println();
    }
```



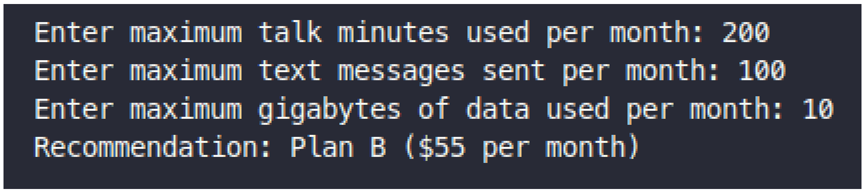
```
Enter a size: 10
* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
--------------------------------------------
1 | 1   2   3   4   5   6   7   8   9   10
2 | 2   4   6   8   10   12   14   16   18   20
3 | 3   6   9   12   15   18   21   24   27   30
4 | 4   8   12   16   20   24   28   32   36   40
5 | 5   10   15   20   25   30   35   40   45   50
6 | 6   12   18   24   30   36   42   48   54   60
7 | 7   14   21   28   35   42   49   56   63   70
8 | 8   16   24   32   40   48   56   64   72   80
9 | 9   18   27   36   45   54   63   72   81   90
10 | 10   20   30   40   50   60   70   80   90   100
```

11. Write a program for Horizon Phones, a provider of cellular phone service. Prompt a user for maximum monthly values for talk minutes used, text messages sent, and gigabytes of data used, and then recommend the best plan for the customer's needs. A customer who needs fewer than 500 minutes of talk and no text or data should accept Plan A at $49 per month. A customer who needs fewer than 500 minutes of talk and any text messages should accept Plan B at $55

per month. A customer who needs 500 or more minutes of talk and no data should accept either Plan C for up to 100 text messages at $61 per month or Plan D for 100 text messages or more at $70 per month. A customer who needs any data should accept Plan E for up to 3 gigabytes at $79 or Plan F for 3 gigabytes or more at $87.



```java
    public static void DataPlan() {
        System.out.print("Enter maximum talk minutes used per month: ");
        int talkMinutes = input.nextInt();
        System.out.print("Enter maximum text messages sent per month: ");
        int textMessages = input.nextInt();
        System.out.print("Enter maximum gigabytes of data used per month:
");
        float gigabytesData = input.nextFloat();
        if (talkMinutes < 500 && textMessages == 0 && gigabytesData == 0) {
            System.out.println("Recommendation: Plan A ($49 per month)");
        } else if (talkMinutes < 500 && textMessages > 0) {
            System.out.println("Recommendation: Plan B ($55 per month)");
        } else if (talkMinutes >= 500 && gigabytesData == 0) {
            if (textMessages < 100) {
                System.out.println("Recommendation: Plan C ($61 per
month)");
            } else {
                System.out.println("Recommendation: Plan D ($70 per
month)");
            }
        } else if (gigabytesData > 0 && gigabytesData <= 3) {
            System.out.println("Recommendation: Plan E ($79 per month)");
        } else if (gigabytesData > 3) {
            System.out.println("Recommendation: Plan F ($87 per month)");
        } else {
            System.out.println("No suitable plan found for the entered
criteria.");
        }
        System.out.println();
    }
```

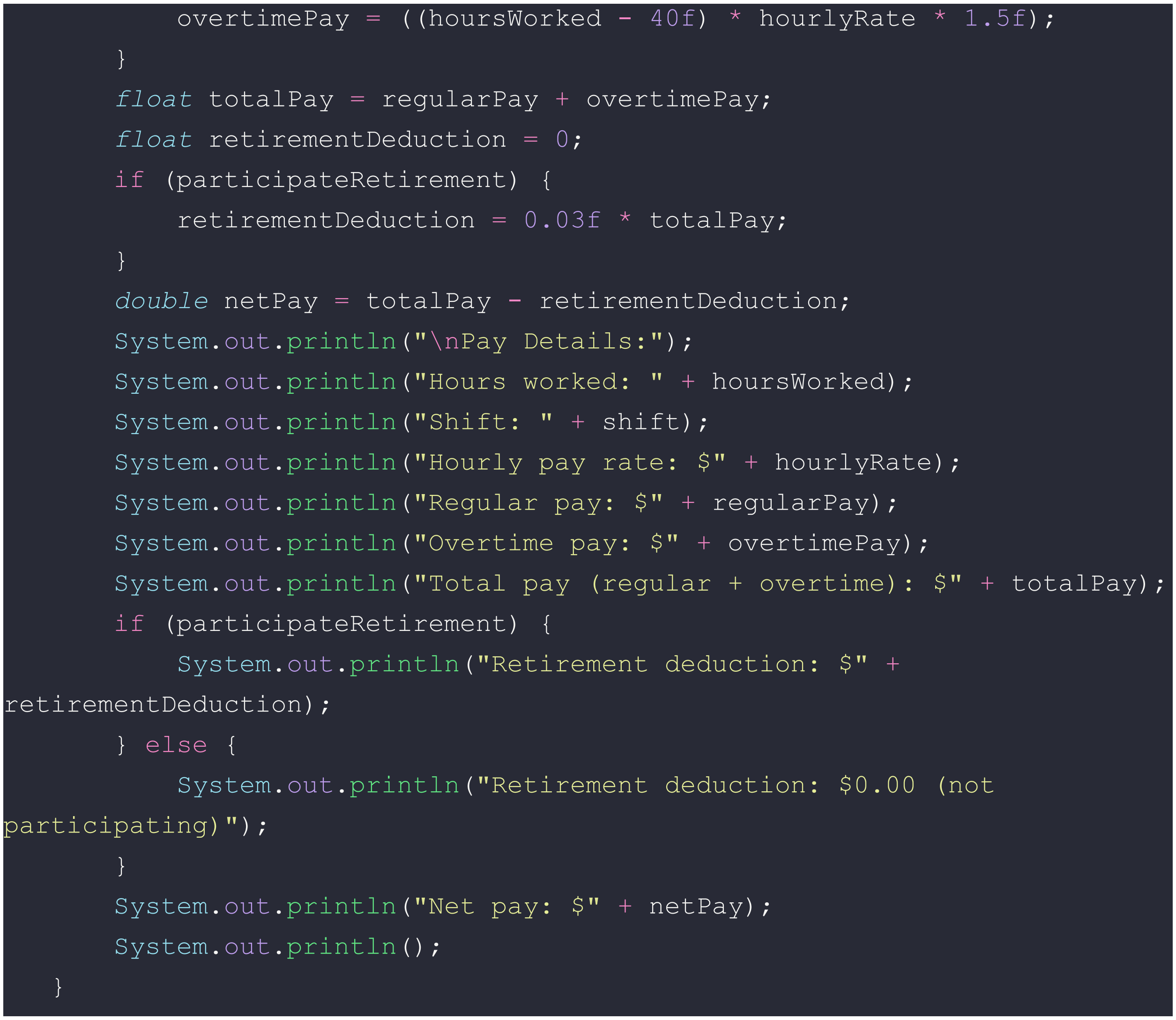


```
Enter maximum talk minutes used per month: 200
Enter maximum text messages sent per month: 100
Enter maximum gigabytes of data used per month: 10
Recommendation: Plan B ($55 per month)
```

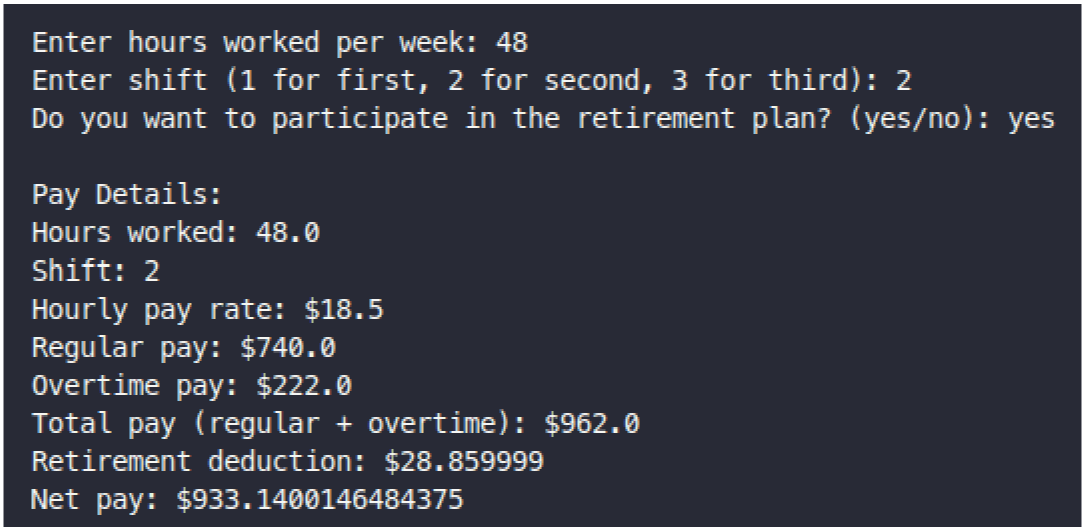
12. Acme Parts runs a small factory and employs workers who are paid one of three hourly rates depending on their shift: first shift, $17 per hour; second shift, $18.50 per hour; third shift, $22 per hour. Each factory worker might work any number of hours per week; any hours greater than 40 are paid at one and one-half times the usual rate. In

addition, second- and third-shift workers can elect to participate in the retirement plan for which 3% of the worker's gross pay is deducted from the paychecks. Write a program that prompts the user for hours worked and shift, and, if the shift is 2 or 3, whether the worker elects the retirement. Display: (1) the hours worked, (2) the shift, (3) the hourly pay rate, (4) the regular pay, (5) overtime pay, (6) the total of regular and overtime pay, and (7) the retirement deduction, if any, and (8) the net pay.

```java
    public static void NetPay() {
        System.out.print("Enter hours worked per week: ");
        float hoursWorked = input.nextFloat();
        System.out.print("Enter shift (1 for first, 2 for second, 3 for third): ");
        int shift = input.nextInt();
        boolean participateRetirement = false;
        if (shift == 2 || shift == 3) {
            System.out.print("Do you want to participate in the retirement plan? (yes/no): ");
            String response = input.next();
            if (response.equalsIgnoreCase("yes")) {
                participateRetirement = true;
            }
        }
        float hourlyRate = 0;
        if (shift == 1) {
            hourlyRate = 17.0f;
        } else if (shift == 2) {
            hourlyRate = 18.50f;
        } else if (shift == 3) {
            hourlyRate = 22.0f;
        } else {
            System.out.println("Invalid shift entered.");
            return;
        }
        float regularPay = 0;
        float overtimePay = 0;
        if (hoursWorked <= 40) {
            regularPay = hoursWorked * hourlyRate;
        } else {
            regularPay = 40 * hourlyRate;
```

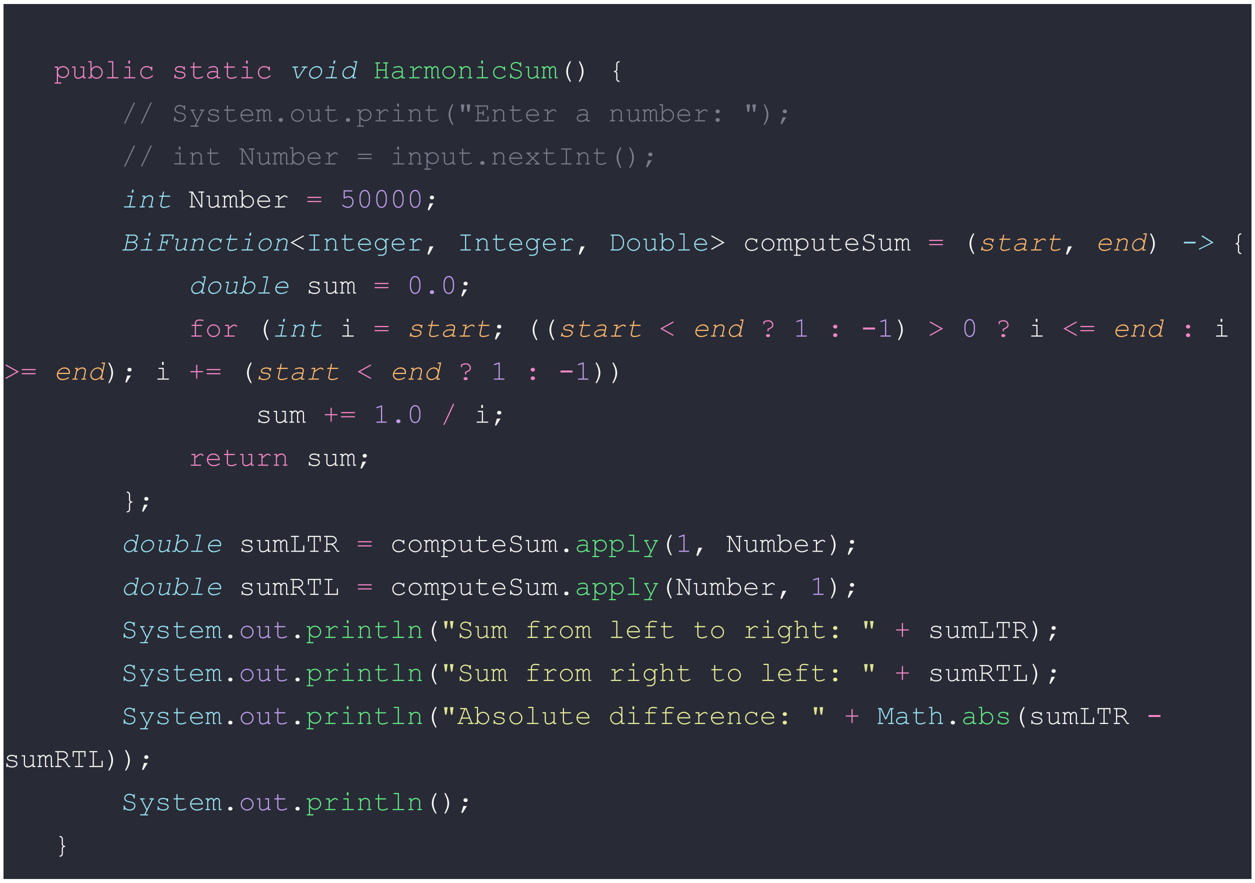


```
            overtimePay = ((hoursWorked - 40f) * hourlyRate * 1.5f);
        }
        float totalPay = regularPay + overtimePay;
        float retirementDeduction = 0;
        if (participateRetirement) {
            retirementDeduction = 0.03f * totalPay;
        }
        double netPay = totalPay - retirementDeduction;
        System.out.println("\nPay Details:");
        System.out.println("Hours worked: " + hoursWorked);
        System.out.println("Shift: " + shift);
        System.out.println("Hourly pay rate: $" + hourlyRate);
        System.out.println("Regular pay: $" + regularPay);
        System.out.println("Overtime pay: $" + overtimePay);
        System.out.println("Total pay (regular + overtime): $" + totalPay);
        if (participateRetirement) {
            System.out.println("Retirement deduction: $" +
retirementDeduction);
        } else {
            System.out.println("Retirement deduction: $0.00 (not
participating)");
        }
        System.out.println("Net pay: $" + netPay);
        System.out.println();
    }
```

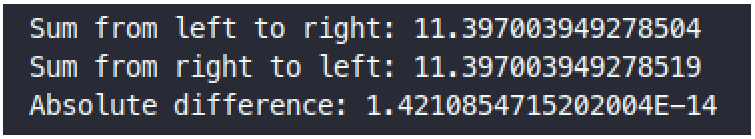


```
Enter hours worked per week: 48
Enter shift (1 for first, 2 for second, 3 for third): 2
Do you want to participate in the retirement plan? (yes/no): yes

Pay Details:
Hours worked: 48.0
Shift: 2
Hourly pay rate: $18.5
Regular pay: $740.0
Overtime pay: $222.0
Total pay (regular + overtime): $962.0
Retirement deduction: $28.859999
Net pay: $933.1400146484375
```

13. Write a program called HarmonicSum to compute the sum of a harmonic series, as shown below, where n=50000. The program shall compute the sum from left-to-right as well as from the right-to-left. Are the two sums the same? Obtain the absolute difference between these two sums and explain the difference. Which sum is more accurate?

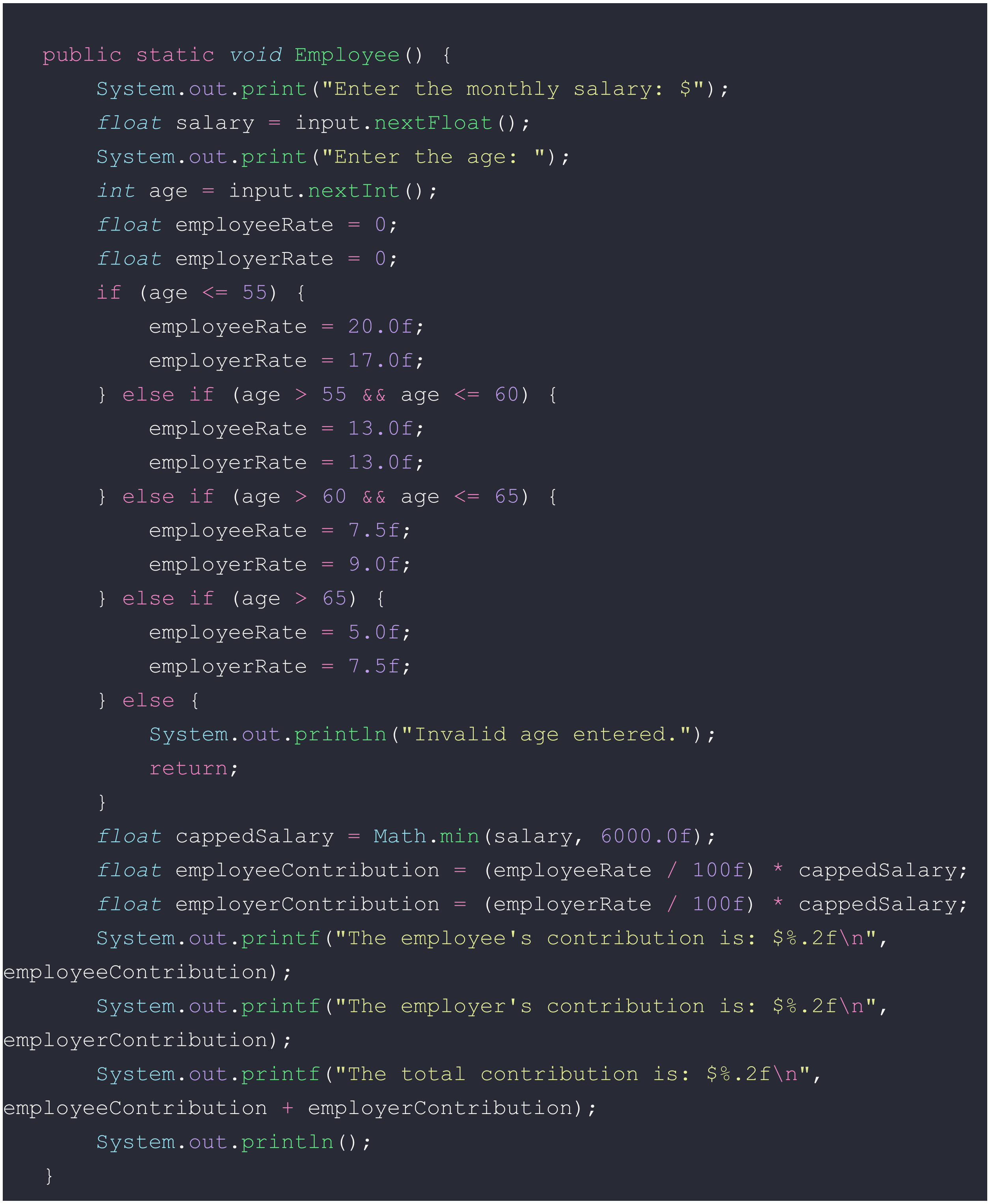


```
public static void HarmonicSum() {
    // System.out.print("Enter a number: ");
    // int Number = input.nextInt();
    int Number = 50000;
    BiFunction<Integer, Integer, Double> computeSum = (start, end) -> {
        double sum = 0.0;
        for (int i = start; ((start < end ? 1 : -1) > 0 ? i <= end : i >= end); i += (start < end ? 1 : -1))
            sum += 1.0 / i;
        return sum;
    };
    double sumLTR = computeSum.apply(1, Number);
    double sumRTL = computeSum.apply(Number, 1);
    System.out.println("Sum from left to right: " + sumLTR);
    System.out.println("Sum from right to left: " + sumRTL);
    System.out.println("Absolute difference: " + Math.abs(sumLTR - sumRTL));
    System.out.println();
}
```

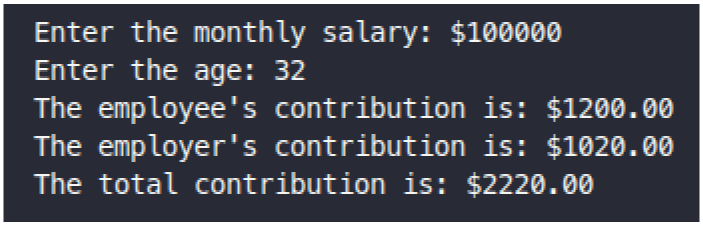


```
Sum from left to right: 11.397003949278504
Sum from right to left: 11.397003949278519
Absolute difference: 1.4210854715202004E-14
```

14. Both the employer and the employee are mandated to contribute a certain percentage of the employee's salary towards the employee's pension fund. The rate is tabulated as follows: However, the contribution is subjected to a salary ceiling of $6,000. In other words, if an employee earns $6,800, only $6,000 attracts employee's and employer's contributions, the remaining $800 does not. Write a program that reads the monthly salary and age of an employee. Your program shall calculate the employee's, employer's and total contributions (in double); and print the results rounded to 2 decimal places. For examples, Enter the monthly salary: $300 Enter the age: 30 The employee's contribution is: $600.00 The employer's contribution is: $510.00 The total contribution is: $1110.00



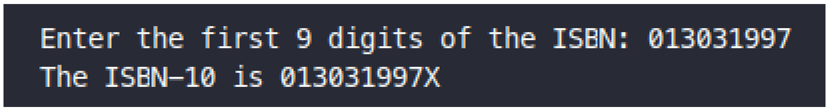
```
   public static void Employee() {
       System.out.print("Enter the monthly salary: $");
       float salary = input.nextFloat();
       System.out.print("Enter the age: ");
       int age = input.nextInt();
       float employeeRate = 0;
       float employerRate = 0;
       if (age <= 55) {
           employeeRate = 20.0f;
           employerRate = 17.0f;
       } else if (age > 55 && age <= 60) {
           employeeRate = 13.0f;
           employerRate = 13.0f;
       } else if (age > 60 && age <= 65) {
           employeeRate = 7.5f;
           employerRate = 9.0f;
       } else if (age > 65) {
           employeeRate = 5.0f;
           employerRate = 7.5f;
       } else {
           System.out.println("Invalid age entered.");
           return;
       }
       float cappedSalary = Math.min(salary, 6000.0f);
       float employeeContribution = (employeeRate / 100f) * cappedSalary;
       float employerContribution = (employerRate / 100f) * cappedSalary;
       System.out.printf("The employee's contribution is: $%.2f\n",
employeeContribution);
       System.out.printf("The employer's contribution is: $%.2f\n",
employerContribution);
       System.out.printf("The total contribution is: $%.2f\n",
employeeContribution + employerContribution);
       System.out.println();
   }
```



```
Enter the monthly salary: $100000
Enter the age: 32
The employee's contribution is: $1200.00
The employer's contribution is: $1020.00
The total contribution is: $2220.00
```

15. An ISBN-10 (International Standard Book Number) consists of 10 digits: d1d2d3d4d5d6d7d8d9d10.The last digit, is a checksum, which is calculated from the other nine digits using the following formula: If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention. Write a program that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros). Your program should read the input as an integer.

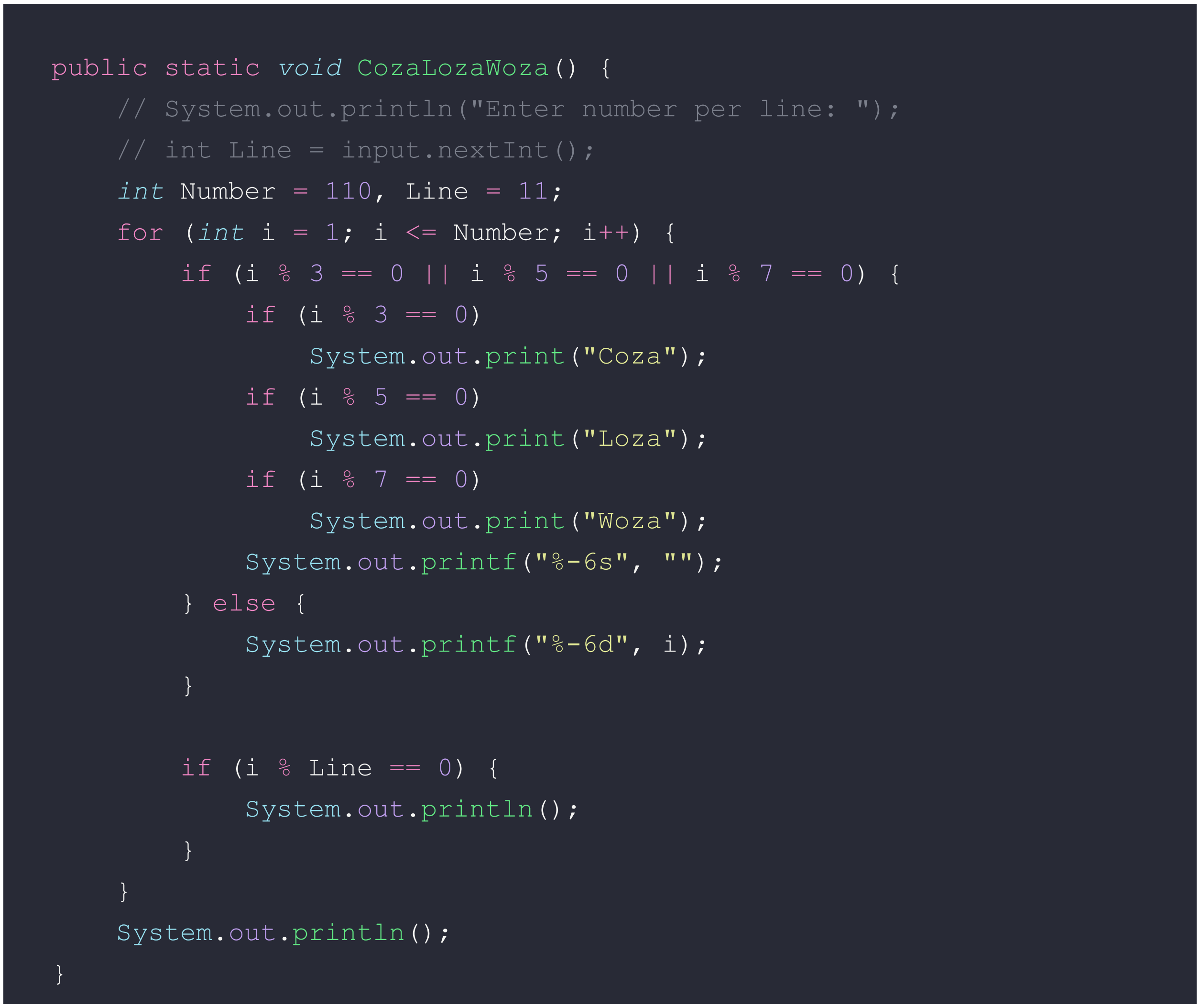
```
   public static void ISBN() {
       input.nextLine(); // clear the buffer
       System.out.print("Enter the first 9 digits of the ISBN: ");
       String first9String = input.nextLine();
       StringBuilder Output = new StringBuilder(first9String);

       Functions = (str, index) -> {
           if (index < 0)
               return 0;
           int digit = str.charAt(index) - '0';
           return (digit * (index + 2)) + Functions.apply(str, index - 1);
       };
       int sum = Functions.apply(Output.reverse(), Output.length() - 1);
       String LastDigit = (((11 - (sum % 11)) == 10) ? "X" :
String.valueOf(11 - (sum % 11)));
       System.out.printf("The ISBN-10 is %09d%s%n",
Integer.valueOf(first9String), LastDigit);
       System.out.println();
   }
```



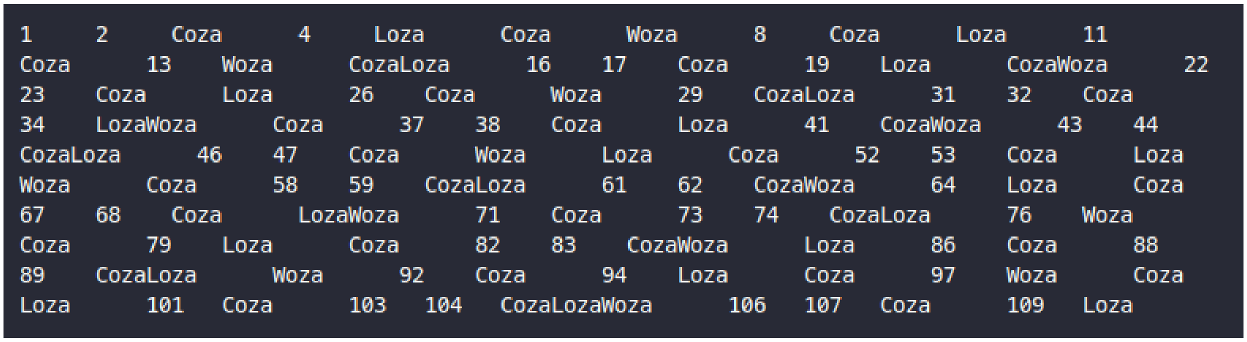
```
Enter the first 9 digits of the ISBN: 013031997
The ISBN-10 is 013031997X
```

16. Write a program called CozaLozaWoza which prints the numbers 1 to 110, 11 numbers per line. The program shall print "Coza" in place of the numbers which are multiples of 3, "Loza" for multiples of 5, "Woza" for multiples of 7, "CozaLoza" for

multiples of 3 and 5, and so on. The output shall look like: 1 2 Coza 4 Loza Coza Woza 8 Coza Loza 11 Coza 13 Woza CozaLoza 16 17 Coza 19 Loza CozaWoza 22 23 Coza Loza 26 Coza Woza 29 CozaLoza 31 32 Coza



```java
public static void CozaLozaWoza() {
    // System.out.println("Enter number per line: ");
    // int Line = input.nextInt();
    int Number = 110, Line = 11;
    for (int i = 1; i <= Number; i++) {
        if (i % 3 == 0 || i % 5 == 0 || i % 7 == 0) {
            if (i % 3 == 0)
                System.out.print("Coza");
            if (i % 5 == 0)
                System.out.print("Loza");
            if (i % 7 == 0)
                System.out.print("Woza");
            System.out.printf("%-6s", "");
        } else {
            System.out.printf("%-6d", i);
        }

        if (i % Line == 0) {
            System.out.println();
        }
    }
    System.out.println();
}
```

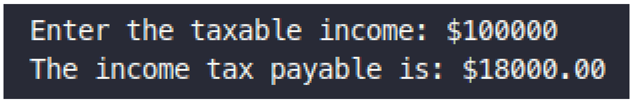


```
1     2     Coza      4     Loza      Coza      Woza      8     Coza      Loza      11
Coza      13    Woza      CozaLoza      16    17    Coza      19    Loza      CozaWoza      22
23    Coza      Loza      26    Coza      Woza      29    CozaLoza      31    32    Coza
34    LozaWoza      Coza      37    38    Coza      Loza      41    CozaWoza      43    44
CozaLoza      46    47    Coza      Woza      Loza      Coza      52    53    Coza      Loza
Woza      Coza      58    59    CozaLoza      61    62    CozaWoza      64    Loza      Coza
67    68    Coza      LozaWoza      71    Coza      73    74    CozaLoza      76    Woza
Coza      79    Loza      Coza      82    83    CozaWoza      Loza      86    Coza      88
89    CozaLoza      Woza      92    Coza      94    Loza      Coza      97    Woza      Coza
Loza      101   Coza      103   104   CozaLozaWoza      106   107   Coza      109   Loza
```

17. The progressive income tax rate is mandated as follows: Taxable Income Rate (%) First $20,000 0 Next $20,000 10 Next $20,000 20 The remaining 30 For example,

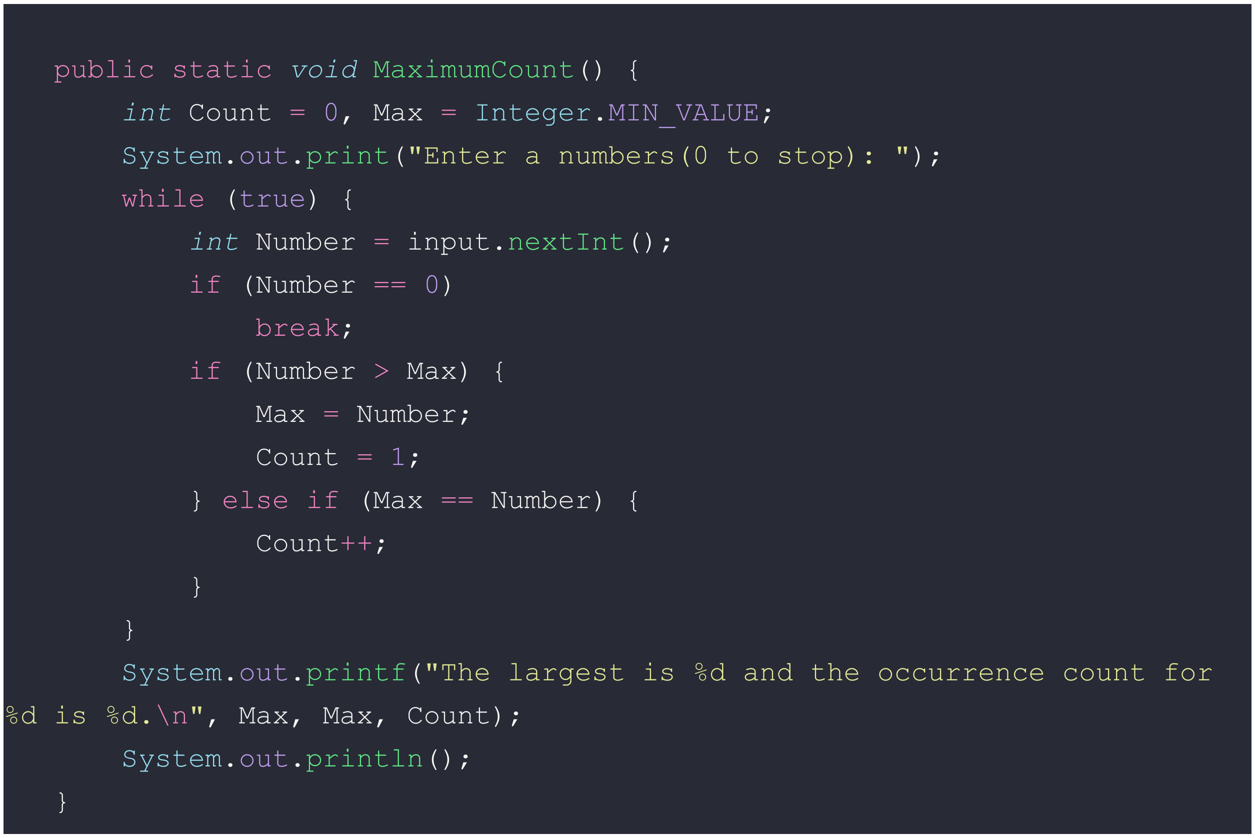
suppose that the taxable income is $85000, the income tax payable is $20000*0% + $20000*10% + $20000*20% + $25000*30%. Write a program that reads the taxable income. The program shall calculate the income tax payable (in double); and print the result rounded to 2 decimal places.

```
  public static void IncomeTax() {
      System.out.print("Enter the taxable income: $");
      float taxableIncome = input.nextFloat();
      float taxPayable = 0;
      if (taxableIncome <= 20000.0f) {
          taxPayable = taxableIncome * 0.0f;
      } else if (taxableIncome <= 40000.0f) {
          taxPayable = 20000.0f * 0.0f + (taxableIncome - 20000.0f) *
0.10f;
      } else if (taxableIncome <= 60000.0f) {
          taxPayable = 20000.0f * 0.0f + (40000.0f - 20000.0f) * 0.10f
                  + (taxableIncome - 40000.0f) * 0.20f;
      } else {
          taxPayable = 20000.0f * 0.0f + (40000.0f - 20000.0f) * 0.10f
                  + (60000.0f - 40000.0f) * 0.20f + (taxableIncome -
60000.0f) * 0.30f;
      }
      System.out.printf("The income tax payable is: $%.2f\n",
taxPayable);
      System.out.println();
  }
```

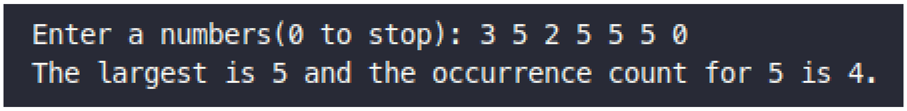


```
Enter the taxable income: $100000
The income tax payable is: $18000.00
```

18. Write a program that reads a list of positive numbers and displays the largest number in the list followed by the number of times it appears using function. The list should be terminated by 0. There is no 0 in the list and there is at least one number before the 0. Do not use an array or non-main method to answer this question. Write your code so that given the same input it would produce the same output like the following example execution: Please type in your numbers -->1 2 3 2 4 5 6 6 0 The largest number is 6 and it appears 2 times

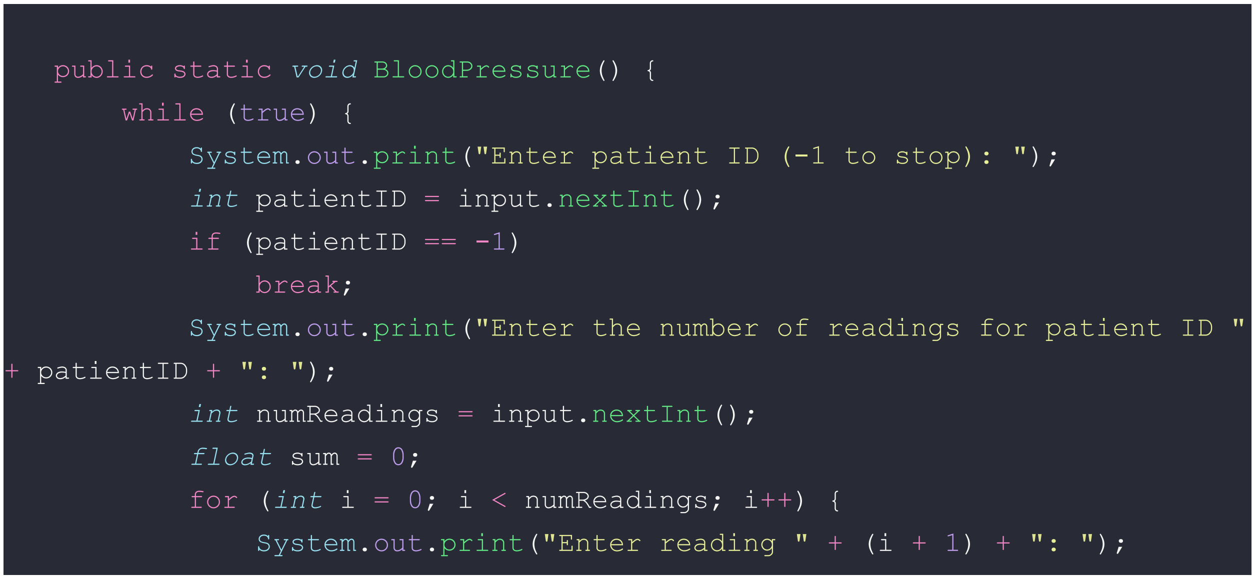


```
   public static void MaximumCount() {
       int Count = 0, Max = Integer.MIN_VALUE;
       System.out.print("Enter a numbers(0 to stop): ");
       while (true) {
           int Number = input.nextInt();
           if (Number == 0)
               break;
           if (Number > Max) {
               Max = Number;
               Count = 1;
           } else if (Max == Number) {
               Count++;
           }
       }
       System.out.printf("The largest is %d and the occurrence count for
%d is %d.\n", Max, Max, Count);
       System.out.println();
   }
```



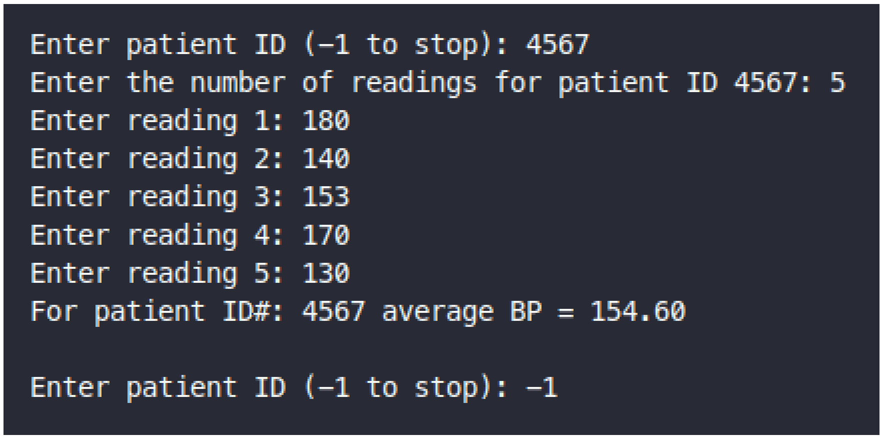
```
Enter a numbers(0 to stop): 3 5 2 5 5 5 0
The largest is 5 and the occurrence count for 5 is 4.
```

19. Print the average blood pressure for different people. Each input has a patient ID, the number of readings for that patient, followed by the actual blood pressure readings. (Use Sentinel-controlled loop)



```
   public static void BloodPressure() {
       while (true) {
           System.out.print("Enter patient ID (-1 to stop): ");
           int patientID = input.nextInt();
           if (patientID == -1)
               break;
           System.out.print("Enter the number of readings for patient ID "
+ patientID + ": ");
           int numReadings = input.nextInt();
           float sum = 0;
           for (int i = 0; i < numReadings; i++) {
               System.out.print("Enter reading " + (i + 1) + ": ");
```

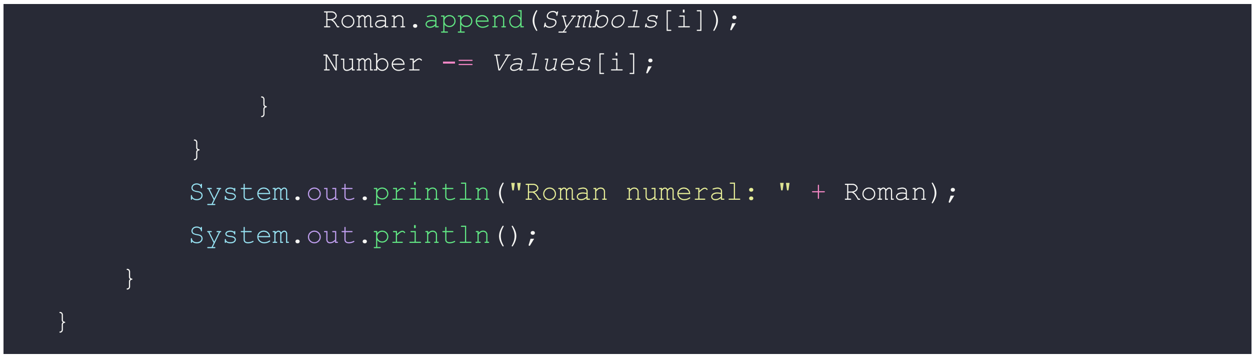
```
                float reading = input.nextFloat();
                sum += reading;
            }
            System.out.printf("For patient ID#: %d average BP = %.2f%n%n",
patientID, sum / numReadings);
        }
        System.out.println();
    }
```



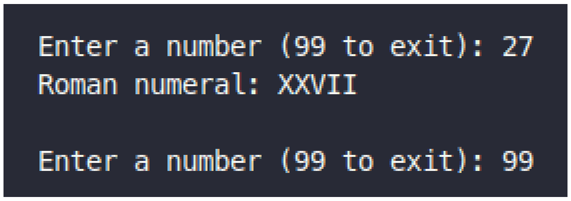
```
Enter patient ID (-1 to stop): 4567
Enter the number of readings for patient ID 4567: 5
Enter reading 1: 180
Enter reading 2: 140
Enter reading 3: 153
Enter reading 4: 170
Enter reading 5: 130
For patient ID#: 4567 average BP = 154.60

Enter patient ID (-1 to stop): -1
```

20. Write an application that allows a user to enter any number that converts it to a roman numeral until the user enters 99. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

```
    public static void IntToRoman() {

        String[] Symbols = { "M", "CM", "D", "CD", "C", "XC", "L", "XL",
"X", "IX", "V", "IV", "I" };
        int[] Values = { 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4,
1 };

        while (true) {
            System.out.print("Enter a number (99 to exit): ");
            int Number = input.nextInt();
            if (Number == 99)
                break;

            StringBuilder Roman = new StringBuilder();
            for (int i = 0; i < Values.length && Number > 0; i++) {
                while (Number >= Values[i]) {
```

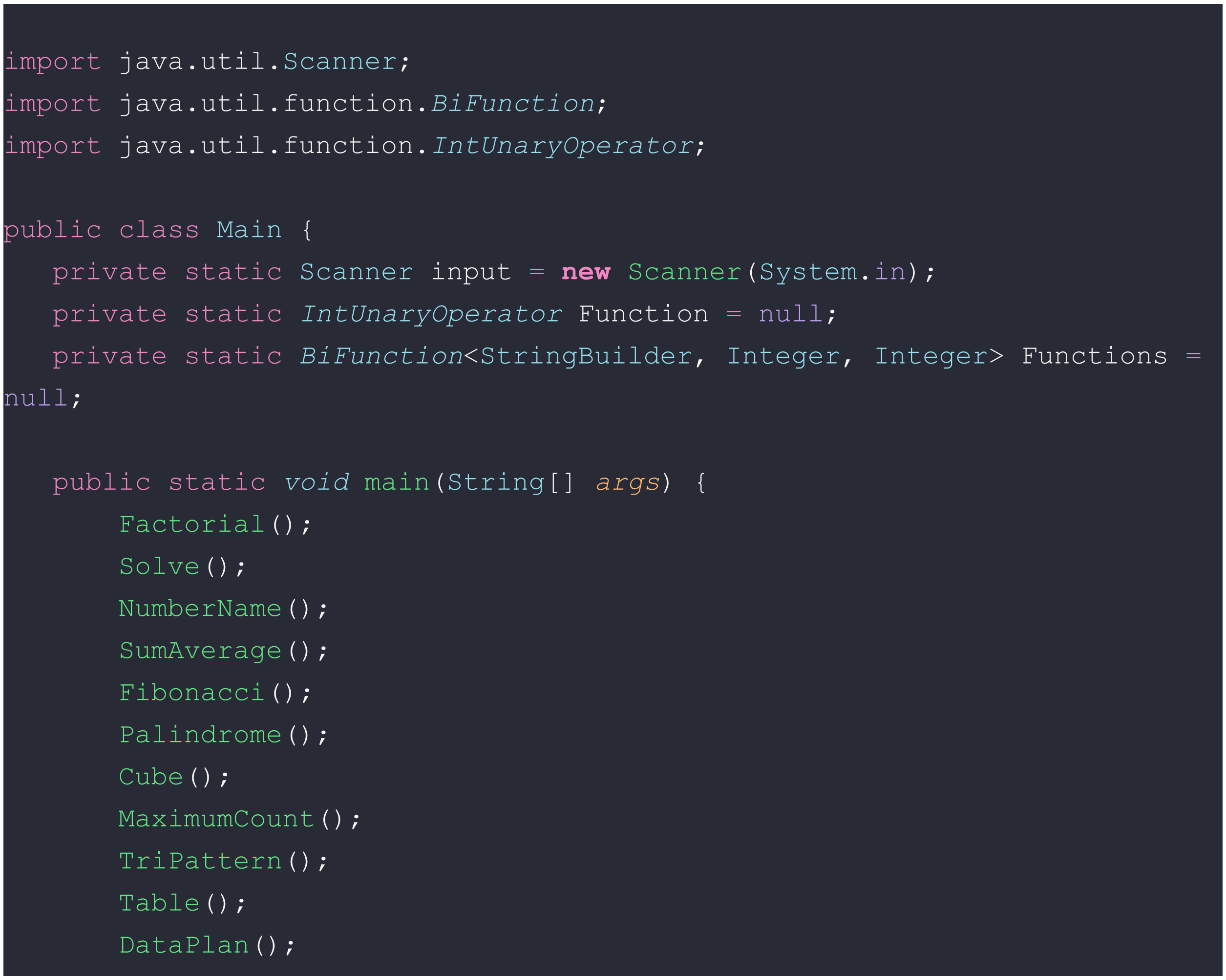


```
                    Roman.append(Symbols[i]);
                    Number -= Values[i];
                }
            }
            System.out.println("Roman numeral: " + Roman);
            System.out.println();
        }
    }
```

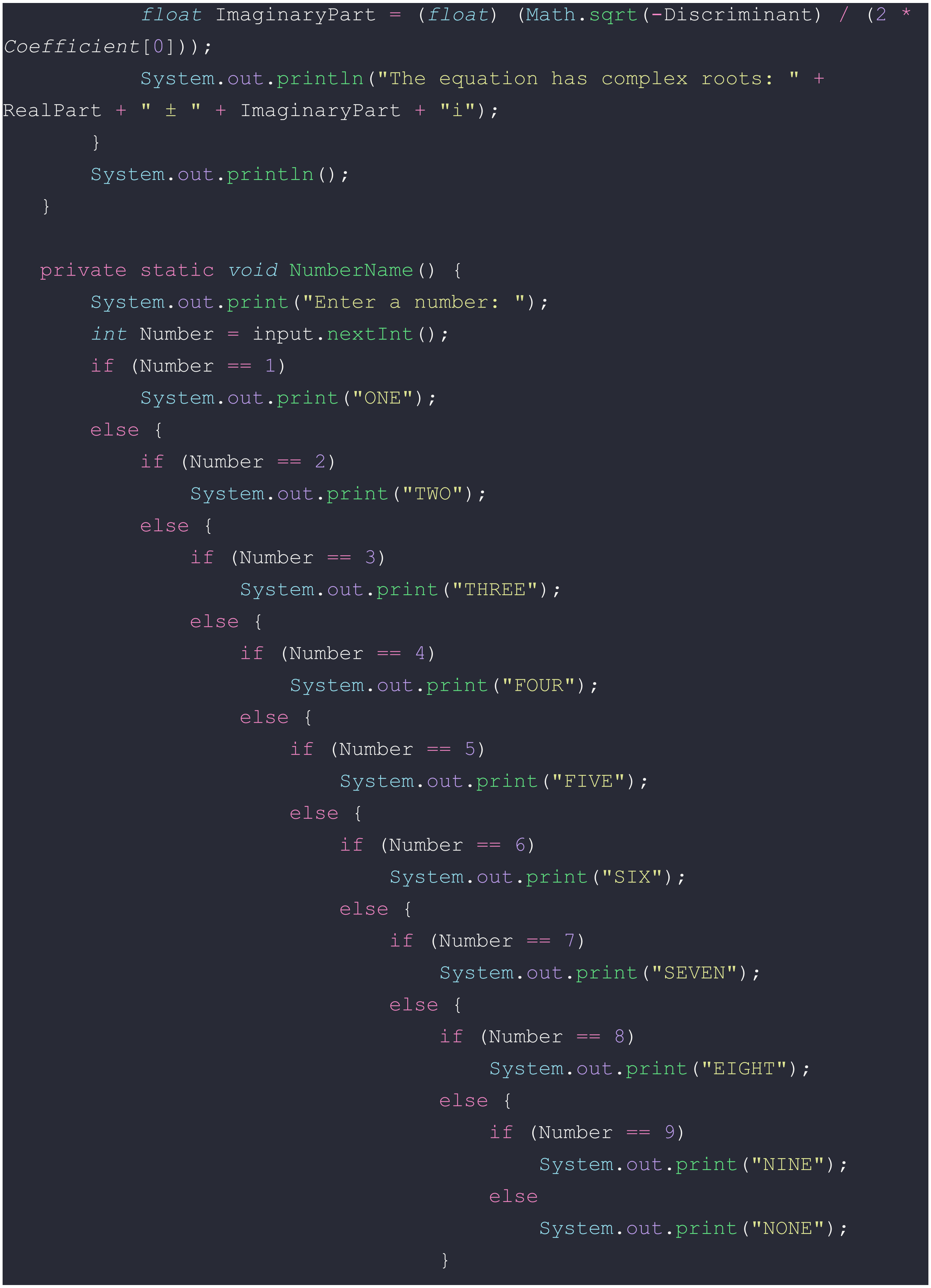


```
Enter a number (99 to exit): 27
Roman numeral: XXVII

Enter a number (99 to exit): 99
```

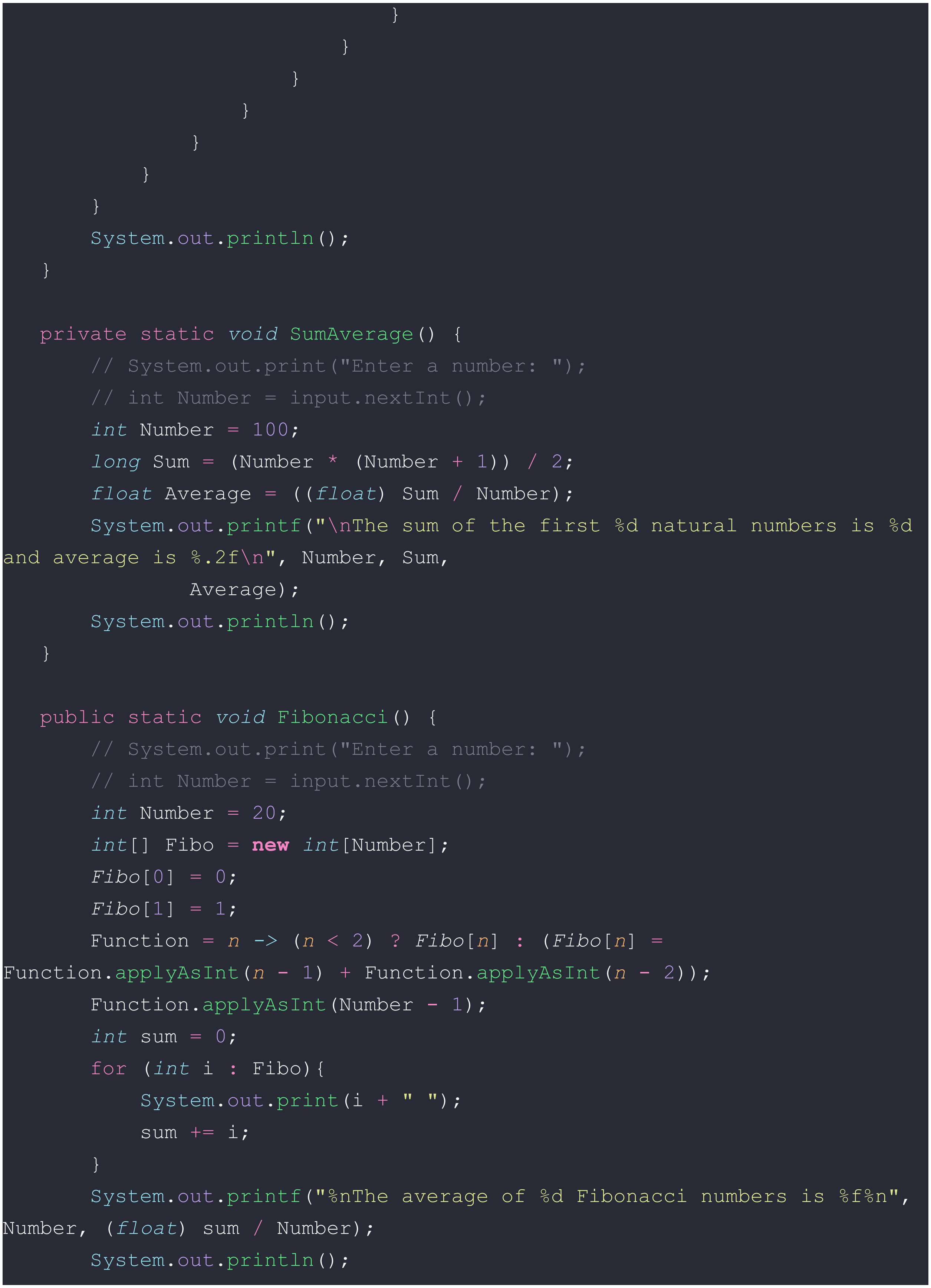
Source code



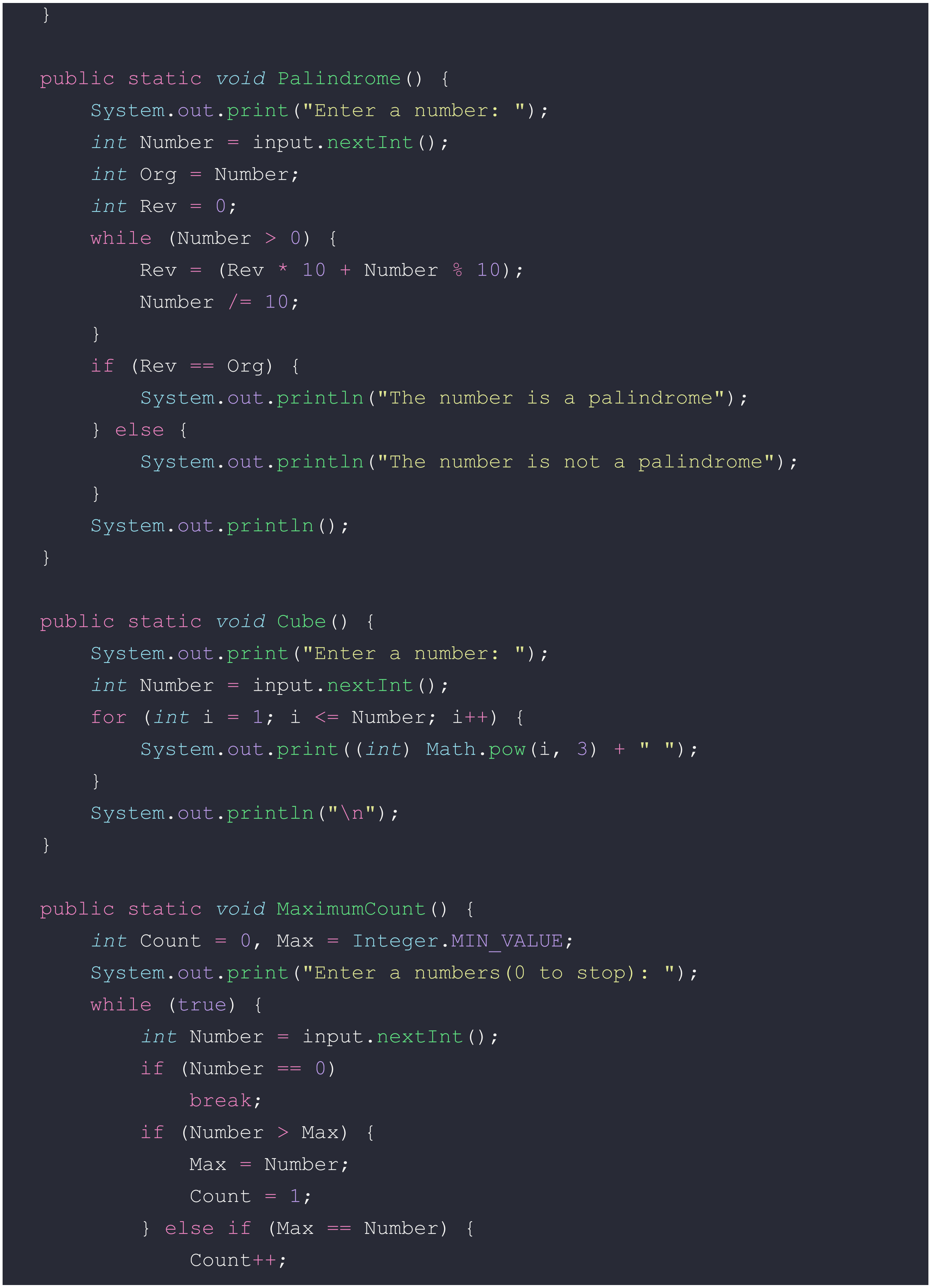
```
import java.util.Scanner;
import java.util.function.BiFunction;
import java.util.function.IntUnaryOperator;

public class Main {
    private static Scanner input = new Scanner(System.in);
    private static IntUnaryOperator Function = null;
    private static BiFunction<StringBuilder, Integer, Integer> Functions =
null;

    public static void main(String[] args) {
        Factorial();
        Solve();
        NumberName();
        SumAverage();
        Fibonacci();
        Palindrome();
        Cube();
        MaximumCount();
        TriPattern();
        Table();
        DataPlan();
        NetPay();
        HarmonicSum();
        Employee();
        ISBN();
        CozaLozaWoza();
        IncomeTax();
        BloodPressure();
        IntToRoman();
    }

    private static void Factorial() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        Function = n -> n == 0 ? 1 : n * Function.applyAsInt(n - 1);
        for (int i = 1; i <= 10; i++)
            System.out.println("Factorial of " + i + " is " +
Function.applyAsInt(i));
        System.out.println();
    }

    private static void Solve() {
        System.out.print("Enter coefficient of A, B, C [in A(x^2) + B(x) +
C]: ");
        float[] Coefficient = { input.nextFloat(), input.nextFloat(),
input.nextFloat() };
        float Discriminant = (Coefficient[1] * Coefficient[1]) - (4 *
Coefficient[0] * Coefficient[2]);
        if (Discriminant > 0) {
            float Root1 = (-Coefficient[1] + (float)
Math.sqrt(Discriminant)) / (2 * Coefficient[0]);
            float Root2 = (-Coefficient[1] - (float)
Math.sqrt(Discriminant)) / (2 * Coefficient[0]);
            System.out.println("The equation has two real and distinct
roots: " + Root1 + " and " + Root2);
        } else if (Discriminant == 0) {
            float Root = -Coefficient[1] / (2 * Coefficient[0]);
            System.out.println("The equation has one real root: " + Root);
        } else {
            float RealPart = -Coefficient[1] / (2 * Coefficient[0]);
```



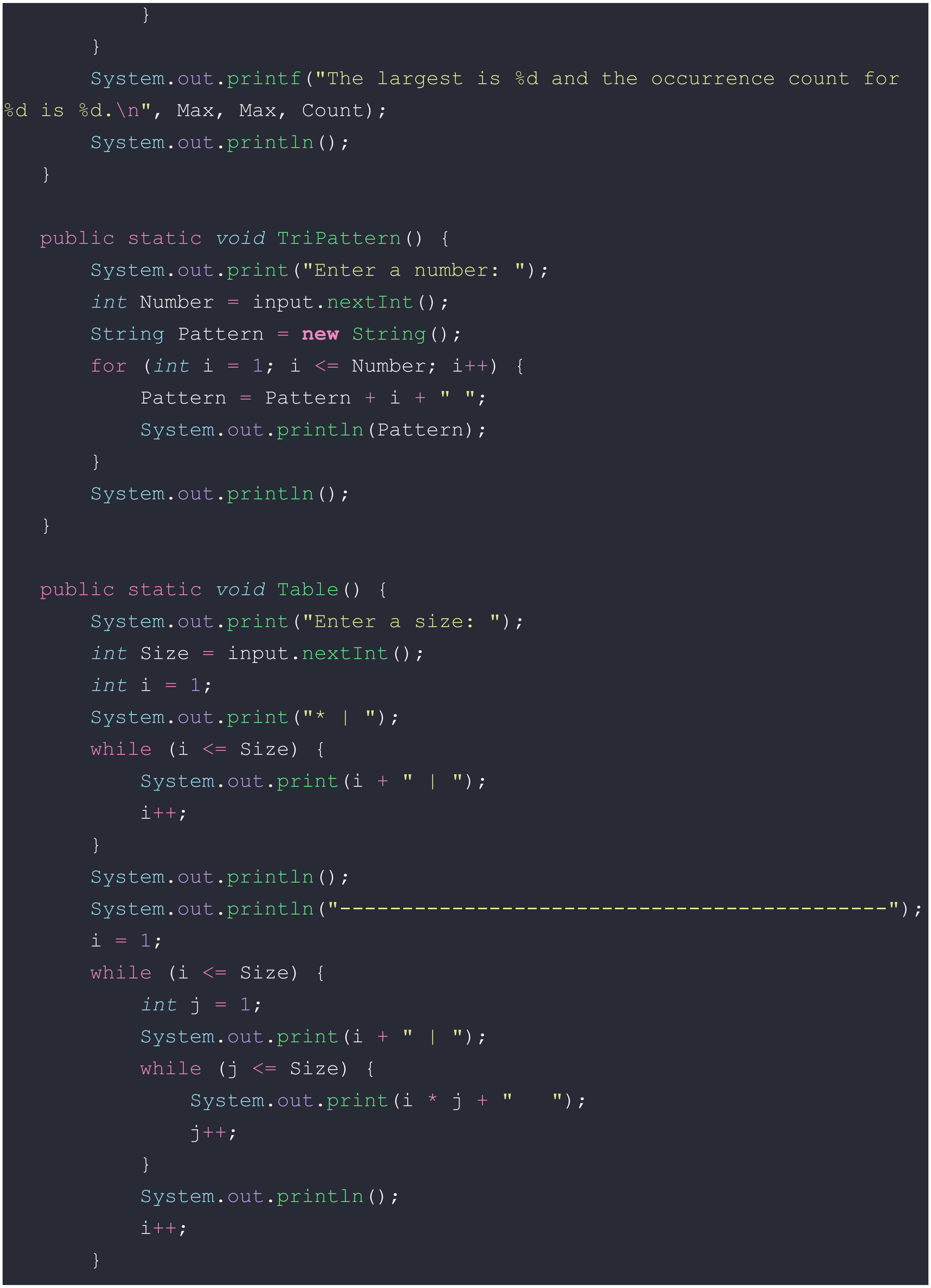
```
            float ImaginaryPart = (float) (Math.sqrt(-Discriminant) / (2 *
Coefficient[0]));
            System.out.println("The equation has complex roots: " +
RealPart + " ± " + ImaginaryPart + "i");
        }
        System.out.println();
    }

    private static void NumberName() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
        if (Number == 1)
            System.out.print("ONE");
        else {
            if (Number == 2)
                System.out.print("TWO");
            else {
                if (Number == 3)
                    System.out.print("THREE");
                else {
                    if (Number == 4)
                        System.out.print("FOUR");
                    else {
                        if (Number == 5)
                            System.out.print("FIVE");
                        else {
                            if (Number == 6)
                                System.out.print("SIX");
                            else {
                                if (Number == 7)
                                    System.out.print("SEVEN");
                                else {
                                    if (Number == 8)
                                        System.out.print("EIGHT");
                                    else {
                                        if (Number == 9)
                                            System.out.print("NINE");
                                        else
                                            System.out.print("NONE");
                                    }
```



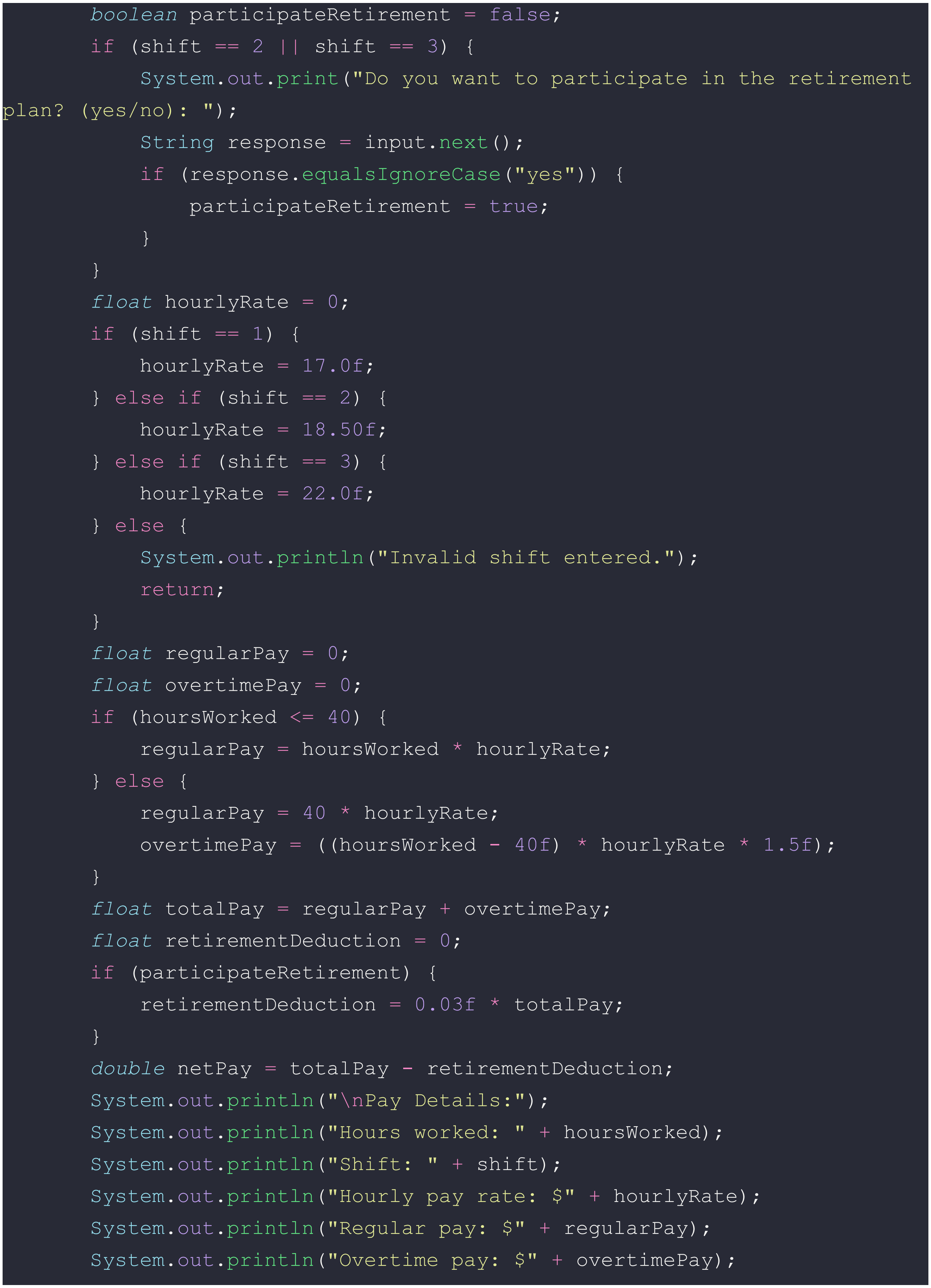
```java
                                }
                            }
                        }
                    }
                }
            }
        }
        System.out.println();
    }

    private static void SumAverage() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        int Number = 100;
        long Sum = (Number * (Number + 1)) / 2;
        float Average = ((float) Sum / Number);
        System.out.printf("\nThe sum of the first %d natural numbers is %d 
and average is %.2f\n", Number, Sum,
                Average);
        System.out.println();
    }

    public static void Fibonacci() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        int Number = 20;
        int[] Fibo = new int[Number];
        Fibo[0] = 0;
        Fibo[1] = 1;
        Function = n -> (n < 2) ? Fibo[n] : (Fibo[n] = 
Function.applyAsInt(n - 1) + Function.applyAsInt(n - 2));
        Function.applyAsInt(Number - 1);
        int sum = 0;
        for (int i : Fibo){
            System.out.print(i + " ");
            sum += i;
        }
        System.out.printf("%nThe average of %d Fibonacci numbers is %f%n", 
Number, (float) sum / Number);
        System.out.println();
```



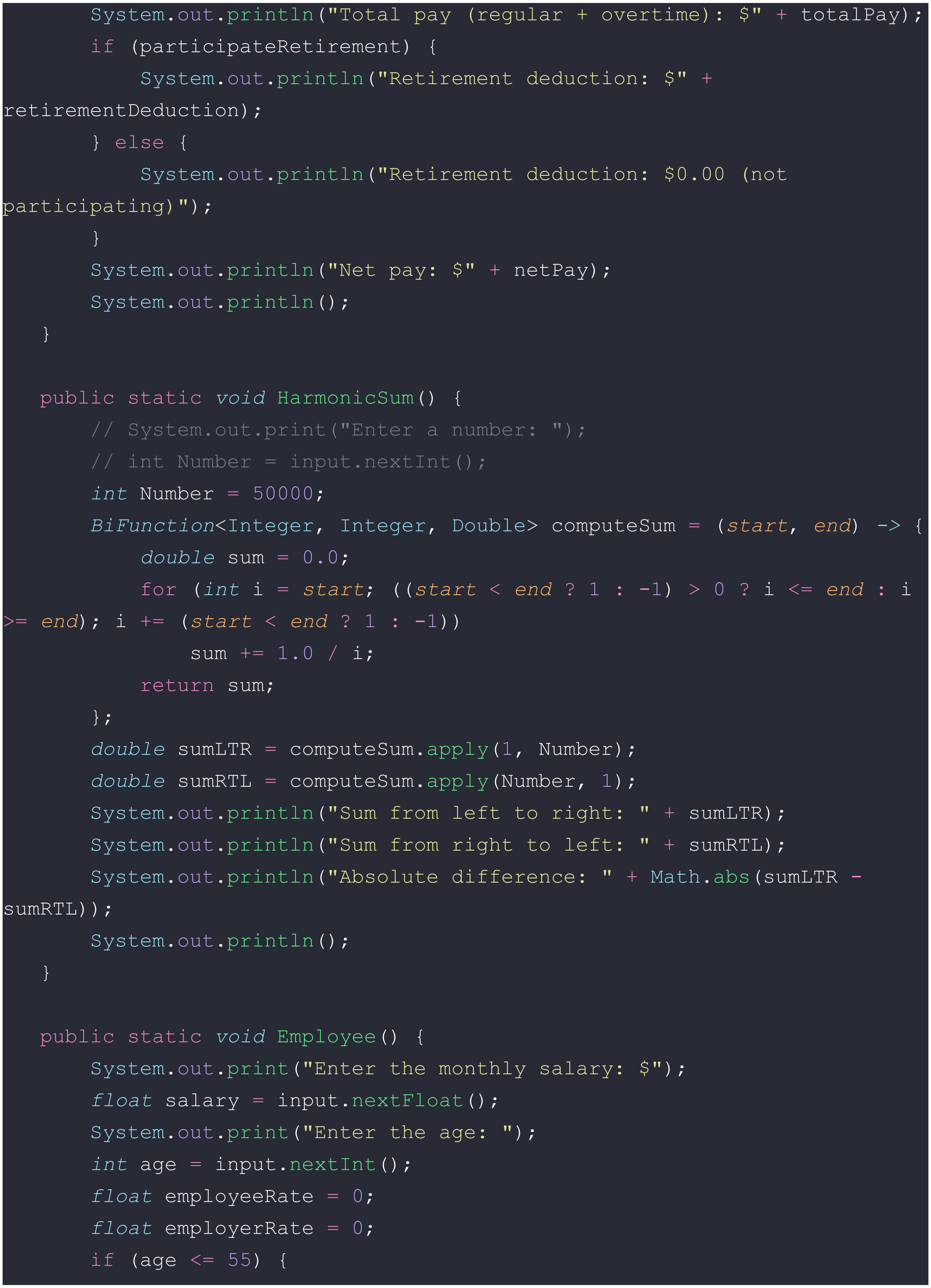
```
    }

    public static void Palindrome() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
        int Org = Number;
        int Rev = 0;
        while (Number > 0) {
            Rev = (Rev * 10 + Number % 10);
            Number /= 10;
        }
        if (Rev == Org) {
            System.out.println("The number is a palindrome");
        } else {
            System.out.println("The number is not a palindrome");
        }
        System.out.println();
    }

    public static void Cube() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
        for (int i = 1; i <= Number; i++) {
            System.out.print((int) Math.pow(i, 3) + " ");
        }
        System.out.println("\n");
    }

    public static void MaximumCount() {
        int Count = 0, Max = Integer.MIN_VALUE;
        System.out.print("Enter a numbers(0 to stop): ");
        while (true) {
            int Number = input.nextInt();
            if (Number == 0)
                break;
            if (Number > Max) {
                Max = Number;
                Count = 1;
            } else if (Max == Number) {
                Count++;
```



```
            }
        }
        System.out.printf("The largest is %d and the occurrence count for
%d is %d.\n", Max, Max, Count);
        System.out.println();
    }

    public static void TriPattern() {
        System.out.print("Enter a number: ");
        int Number = input.nextInt();
        String Pattern = new String();
        for (int i = 1; i <= Number; i++) {
            Pattern = Pattern + i + " ";
            System.out.println(Pattern);
        }
        System.out.println();
    }

    public static void Table() {
        System.out.print("Enter a size: ");
        int Size = input.nextInt();
        int i = 1;
        System.out.print("* | ");
        while (i <= Size) {
            System.out.print(i + " | ");
            i++;
        }
        System.out.println();
        System.out.println("--------------------------------------------");
        i = 1;
        while (i <= Size) {
            int j = 1;
            System.out.print(i + " | ");
            while (j <= Size) {
                System.out.print(i * j + "   ");
                j++;
            }
            System.out.println();
            i++;
        }
        System.out.println();
    }

    public static void DataPlan() {
        System.out.print("Enter maximum talk minutes used per month: ");
        int talkMinutes = input.nextInt();
        System.out.print("Enter maximum text messages sent per month: ");
        int textMessages = input.nextInt();
        System.out.print("Enter maximum gigabytes of data used per month:
");
        float gigabytesData = input.nextFloat();
        if (talkMinutes < 500 && textMessages == 0 && gigabytesData == 0) {
            System.out.println("Recommendation: Plan A ($49 per month)");
        } else if (talkMinutes < 500 && textMessages > 0) {
            System.out.println("Recommendation: Plan B ($55 per month)");
        } else if (talkMinutes >= 500 && gigabytesData == 0) {
            if (textMessages < 100) {
                System.out.println("Recommendation: Plan C ($61 per
month)");
            } else {
                System.out.println("Recommendation: Plan D ($70 per
month)");
            }
        } else if (gigabytesData > 0 && gigabytesData <= 3) {
            System.out.println("Recommendation: Plan E ($79 per month)");
        } else if (gigabytesData > 3) {
            System.out.println("Recommendation: Plan F ($87 per month)");
        } else {
            System.out.println("No suitable plan found for the entered
criteria.");
        }
        System.out.println();
    }

    public static void NetPay() {
        System.out.print("Enter hours worked per week: ");
        float hoursWorked = input.nextFloat();
        System.out.print("Enter shift (1 for first, 2 for second, 3 for
third): ");
        int shift = input.nextInt();
```

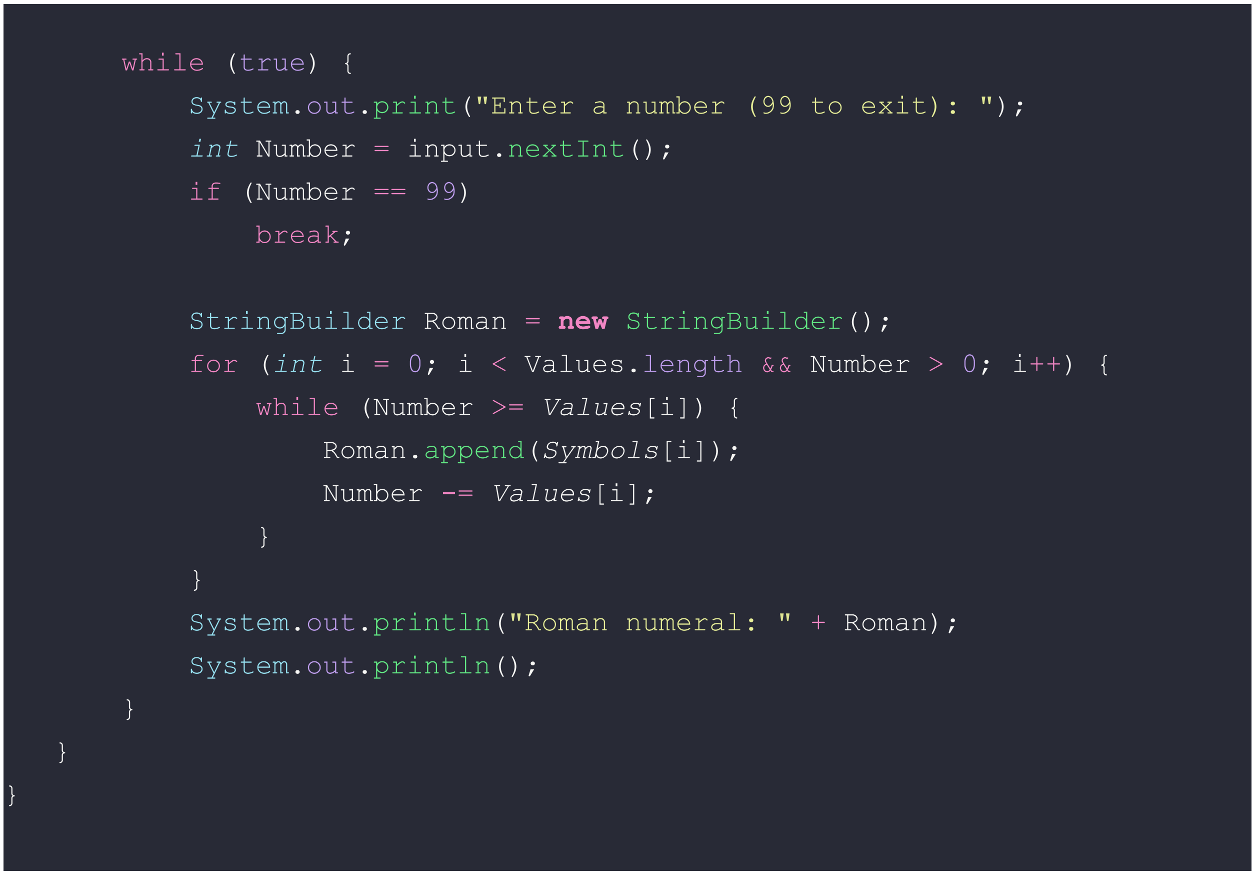


```
        boolean participateRetirement = false;
        if (shift == 2 || shift == 3) {
            System.out.print("Do you want to participate in the retirement
plan? (yes/no): ");
            String response = input.next();
            if (response.equalsIgnoreCase("yes")) {
                participateRetirement = true;
            }
        }
        float hourlyRate = 0;
        if (shift == 1) {
            hourlyRate = 17.0f;
        } else if (shift == 2) {
            hourlyRate = 18.50f;
        } else if (shift == 3) {
            hourlyRate = 22.0f;
        } else {
            System.out.println("Invalid shift entered.");
            return;
        }
        float regularPay = 0;
        float overtimePay = 0;
        if (hoursWorked <= 40) {
            regularPay = hoursWorked * hourlyRate;
        } else {
            regularPay = 40 * hourlyRate;
            overtimePay = ((hoursWorked - 40f) * hourlyRate * 1.5f);
        }
        float totalPay = regularPay + overtimePay;
        float retirementDeduction = 0;
        if (participateRetirement) {
            retirementDeduction = 0.03f * totalPay;
        }
        double netPay = totalPay - retirementDeduction;
        System.out.println("\nPay Details:");
        System.out.println("Hours worked: " + hoursWorked);
        System.out.println("Shift: " + shift);
        System.out.println("Hourly pay rate: $" + hourlyRate);
        System.out.println("Regular pay: $" + regularPay);
        System.out.println("Overtime pay: $" + overtimePay);
```



```
        System.out.println("Total pay (regular + overtime): $" + totalPay);
        if (participateRetirement) {
            System.out.println("Retirement deduction: $" +
retirementDeduction);
        } else {
            System.out.println("Retirement deduction: $0.00 (not
participating)");
        }
        System.out.println("Net pay: $" + netPay);
        System.out.println();
    }

    public static void HarmonicSum() {
        // System.out.print("Enter a number: ");
        // int Number = input.nextInt();
        int Number = 50000;
        BiFunction<Integer, Integer, Double> computeSum = (start, end) -> {
            double sum = 0.0;
            for (int i = start; ((start < end ? 1 : -1) > 0 ? i <= end : i
>= end); i += (start < end ? 1 : -1))
                sum += 1.0 / i;
            return sum;
        };
        double sumLTR = computeSum.apply(1, Number);
        double sumRTL = computeSum.apply(Number, 1);
        System.out.println("Sum from left to right: " + sumLTR);
        System.out.println("Sum from right to left: " + sumRTL);
        System.out.println("Absolute difference: " + Math.abs(sumLTR -
sumRTL));
        System.out.println();
    }

    public static void Employee() {
        System.out.print("Enter the monthly salary: $");
        float salary = input.nextFloat();
        System.out.print("Enter the age: ");
        int age = input.nextInt();
        float employeeRate = 0;
        float employerRate = 0;
        if (age <= 55) {
            employeeRate = 20.0f;
            employerRate = 17.0f;
        } else if (age > 55 && age <= 60) {
            employeeRate = 13.0f;
            employerRate = 13.0f;
        } else if (age > 60 && age <= 65) {
            employeeRate = 7.5f;
            employerRate = 9.0f;
        } else if (age > 65) {
            employeeRate = 5.0f;
            employerRate = 7.5f;
        } else {
            System.out.println("Invalid age entered.");
            return;
        }
        float cappedSalary = Math.min(salary, 6000.0f);
        float employeeContribution = (employeeRate / 100f) * cappedSalary;
        float employerContribution = (employerRate / 100f) * cappedSalary;
        System.out.printf("The employee's contribution is: $%.2f\n",
employeeContribution);
        System.out.printf("The employer's contribution is: $%.2f\n",
employerContribution);
        System.out.printf("The total contribution is: $%.2f\n",
employeeContribution + employerContribution);
        System.out.println();
    }

    public static void ISBN() {
        input.nextLine(); // clear the buffer
        System.out.print("Enter the first 9 digits of the ISBN: ");
        String first9String = input.nextLine();
        StringBuilder Output = new StringBuilder(first9String);

        Functions = (str, index) -> {
            if (index < 0)
                return 0;
            int digit = str.charAt(index) - '0';
            return (digit * (index + 2)) + Functions.apply(str, index - 1);
        };
        int sum = Functions.apply(Output.reverse(), Output.length() - 1);
        String LastDigit = (((11 - (sum % 11)) == 10) ? "X" :
String.valueOf(11 - (sum % 11)));
        System.out.printf("The ISBN-10 is %09d%s%n",
Integer.valueOf(first9String), LastDigit);
        System.out.println();
    }

    public static void CozaLozaWoza() {
        // System.out.println("Enter number per line: ");
        // int Line = input.nextInt();
        int Number = 110, Line = 11;
        for (int i = 1; i <= Number; i++) {
            if (i % 3 == 0 || i % 5 == 0 || i % 7 == 0) {
                if (i % 3 == 0)
                    System.out.print("Coza");
                if (i % 5 == 0)
                    System.out.print("Loza");
                if (i % 7 == 0)
                    System.out.print("Woza");
                System.out.printf("%-6s", "");
            } else {
                System.out.printf("%-6d", i);
            }

            if (i % Line == 0) {
                System.out.println();
            }
        }
        System.out.println();
    }

    public static void IncomeTax() {
        System.out.print("Enter the taxable income: $");
        float taxableIncome = input.nextFloat();
        float taxPayable = 0;
        if (taxableIncome <= 20000.0f) {
            taxPayable = taxableIncome * 0.0f;
        } else if (taxableIncome <= 40000.0f) {
            taxPayable = 20000.0f * 0.0f + (taxableIncome - 20000.0f) *
0.10f;
```

```
       } else if (taxableIncome <= 60000.0f) {
           taxPayable = 20000.0f * 0.0f + (40000.0f - 20000.0f) * 0.10f
                   + (taxableIncome - 40000.0f) * 0.20f;
       } else {
           taxPayable = 20000.0f * 0.0f + (40000.0f - 20000.0f) * 0.10f
                   + (60000.0f - 40000.0f) * 0.20f + (taxableIncome -
60000.0f) * 0.30f;
       }
       System.out.printf("The income tax payable is: $%.2f\n",
taxPayable);
       System.out.println();
   }

   public static void BloodPressure() {
       while (true) {
           System.out.print("Enter patient ID (-1 to stop): ");
           int patientID = input.nextInt();
           if (patientID == -1)
               break;
           System.out.print("Enter the number of readings for patient ID "
+ patientID + ": ");
           int numReadings = input.nextInt();
           float sum = 0;
           for (int i = 0; i < numReadings; i++) {
               System.out.print("Enter reading " + (i + 1) + ": ");
               float reading = input.nextFloat();
               sum += reading;
           }
           System.out.printf("For patient ID#: %d average BP = %.2f%n%n",
patientID, sum / numReadings);
       }
       System.out.println();
   }

   public static void IntToRoman() {

       String[] Symbols = { "M", "CM", "D", "CD", "C", "XC", "L", "XL",
"X", "IX", "V", "IV", "I" };
       int[] Values = { 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4,
1 };
```



```
        while (true) {
            System.out.print("Enter a number (99 to exit): ");
            int Number = input.nextInt();
            if (Number == 99)
                break;

            StringBuilder Roman = new StringBuilder();
            for (int i = 0; i < Values.length && Number > 0; i++) {
                while (Number >= Values[i]) {
                    Roman.append(Symbols[i]);
                    Number -= Values[i];
                }
            }
            System.out.println("Roman numeral: " + Roman);
            System.out.println();
        }
    }
}
```