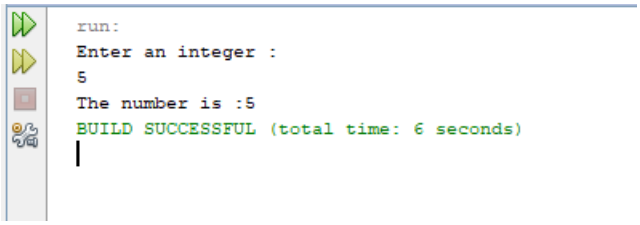
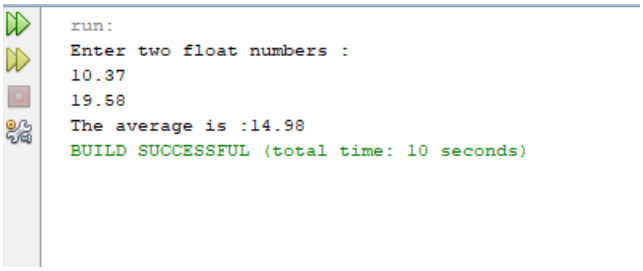
1a) Write a Java program that prompts the user to enter an integer, reads the input, and displays the entered integer on the console.

```
package basics;

import java.util.*;

public class Basics {

public static void main(String[] args) {

    Scanner scan = new Scanner (System.in);

    System.out.println("Enter an integer :");

    int a = scan.nextInt();

    System.out.println("The number is :"+ a);

 }

}
```



b) Develop a Java program that reads two floating-point numbers from the user, calculates their average, and displays the result on the console with two decimal places.

```
import java.util.*;

public class Basics {

public static void main(String[] args) {

    Scanner scan = new Scanner (System.in);

    System.out.println("Enter two float numbers :");

    float a = scan.nextFloat();

    float b = scan.nextFloat();

    float avg = (a+b)/2;

    System.out.format("The average is :%.2f",avg);

    System.out.println();

 }

}
```



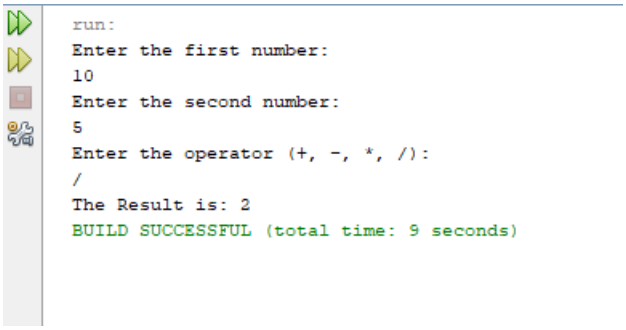
2. Implement a Java program that simulates a basic calculator with functionalities to perform addition, subtraction, multiplication, and division.

The program should prompt the user to enter two numbers and an operator (+, -, , /), perform the corresponding operation, and display the result.

Ensure to handle division by zero and invalid operator inputs.

```
package basics;

import java.util.Scanner;

public class Basics {

   public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);

      System.out.println("Enter the first number:");
      int num1 = scan.nextInt();

      System.out.println("Enter the second number:");
      int num2 = scan.nextInt();

      char operator;

       System.out.println("Enter the operator (+, -, *, /):");
       operator = scan.next().charAt(0);


       int result = performOperation(num1, num2, operator);
       System.out.println("The Result is: " + result);
       scan.close();
   }

   private static int performOperation(int num1, int num2, char operator) {
      switch (operator) {
         case '+':
            int add = num1 + num2;
            return add;
         case '-':
            int subtract = num1 - num2;
            return subtract;
         case '*':
            int multiply = num1*num2;
            return multiply;
```

```
        case '/':
          int divide = num1/num2;
          if(num2==0)
            throw new ArithmeticException("Divide by zero Error");
          else
            return divide;
        default:
            throw new IllegalArgumentException("Invalid operator");
      }
    }
}
```



```
run:
Enter the first number:
10
Enter the second number:
5
Enter the operator (+, -, *, /):
/
The Result is: 2
BUILD SUCCESSFUL (total time: 9 seconds)
```

3. Write an Java program to determine if a number n is happy.

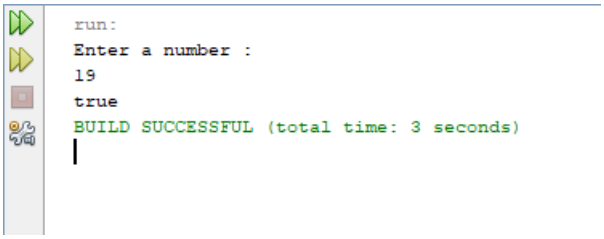
A happy number is a number defined by the following process:

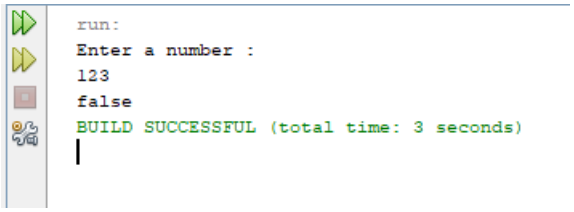
Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1.Those numbers for which this process ends in 1 are happy.

Print true *if* n *is a happy number, and* false *if not*

```
package placement1;

import java.util.Scanner;

public class Placement1 {

public static void main(String[] args) {

    int number ;

    Scanner scan = new Scanner(System.in);

    System.out.println("Enter a number :");

    number = scan.nextInt();

    if ( number > 0)

    {

    boolean isHappy = isHappyNumber(number);
```

```
        System.out.println(isHappy);
        }
        else
        {
            System.out.println("Invalid number !! integer should be positive");
        }
       scan.close();
    }
  private static boolean isHappyNumber(int n) {
        int current = n;
        int next = n;
        do {
            current = getNextSumOfSquares(current);
            next = getNextSumOfSquares(getNextSumOfSquares(next));
        } while (current != next);
      return current == 1;
    }
  private static int getNextSumOfSquares(int n) {
        int sum = 0;
        while (n > 0) {
            int digit = n % 10;
            sum += digit * digit;
            n /= 10;
        }
        return sum;
    }
}
```


run:
Enter a number :
19
true
BUILD SUCCESSFUL (total time: 3 seconds)


run:
Enter a number :
123
false
BUILD SUCCESSFUL (total time: 3 seconds)