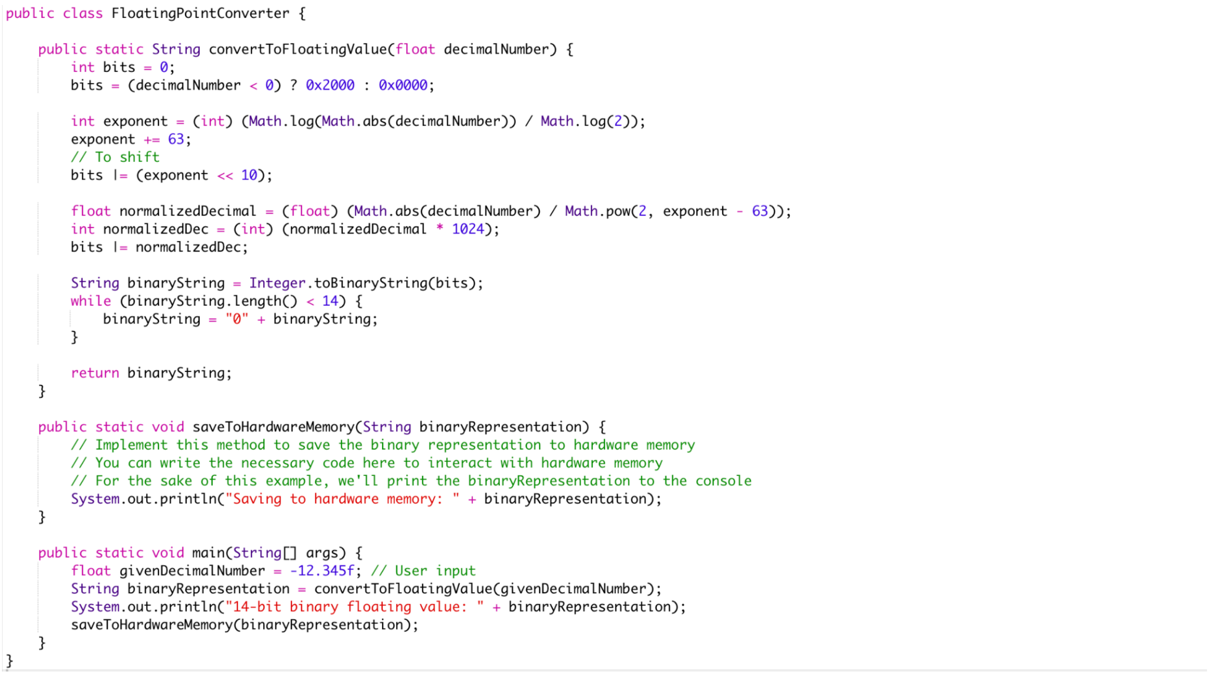
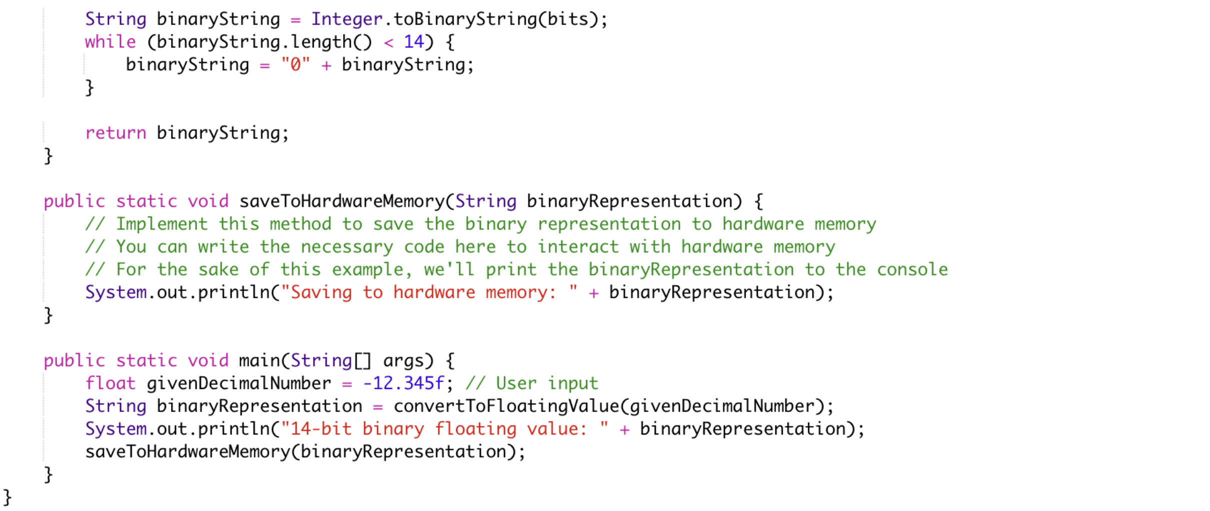
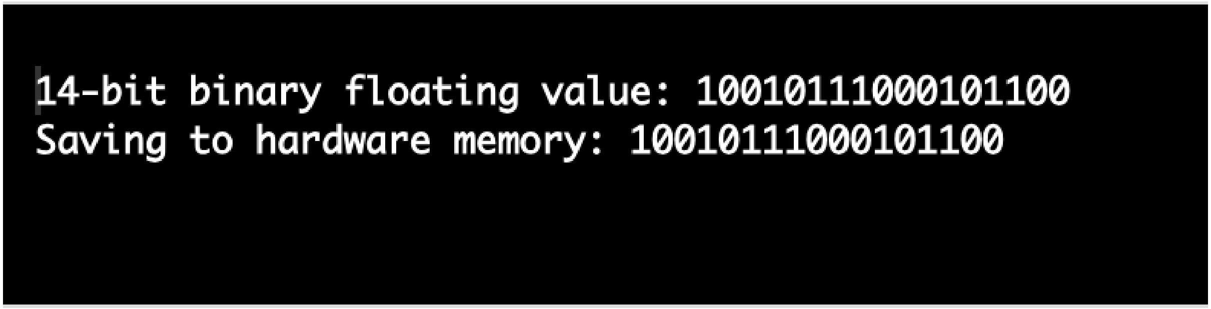
Question 1



```java
public class FloatingPointConverter {

    public static String convertToFloatingValue(float decimalNumber) {
        int bits = 0;
        bits = (decimalNumber < 0) ? 0x2000 : 0x0000;

        int exponent = (int) (Math.log(Math.abs(decimalNumber)) / Math.log(2));
        exponent += 63;
        // To shift
        bits |= (exponent << 10);

        float normalizedDecimal = (float) (Math.abs(decimalNumber) / Math.pow(2, exponent - 63));
        int normalizedDec = (int) (normalizedDecimal * 1024);
        bits |= normalizedDec;

        String binaryString = Integer.toBinaryString(bits);
        while (binaryString.length() < 14) {
            binaryString = "0" + binaryString;
        }

        return binaryString;
    }

    public static void saveToHardwareMemory(String binaryRepresentation) {
        // Implement this method to save the binary representation to hardware memory
        // You can write the necessary code here to interact with hardware memory
        // For the sake of this example, we'll print the binaryRepresentation to the console
        System.out.println("Saving to hardware memory: " + binaryRepresentation);
    }

    public static void main(String[] args) {
        float givenDecimalNumber = -12.345f; // User input
        String binaryRepresentation = convertToFloatingValue(givenDecimalNumber);
        System.out.println("14-bit binary floating value: " + binaryRepresentation);
        saveToHardwareMemory(binaryRepresentation);
    }
}
```



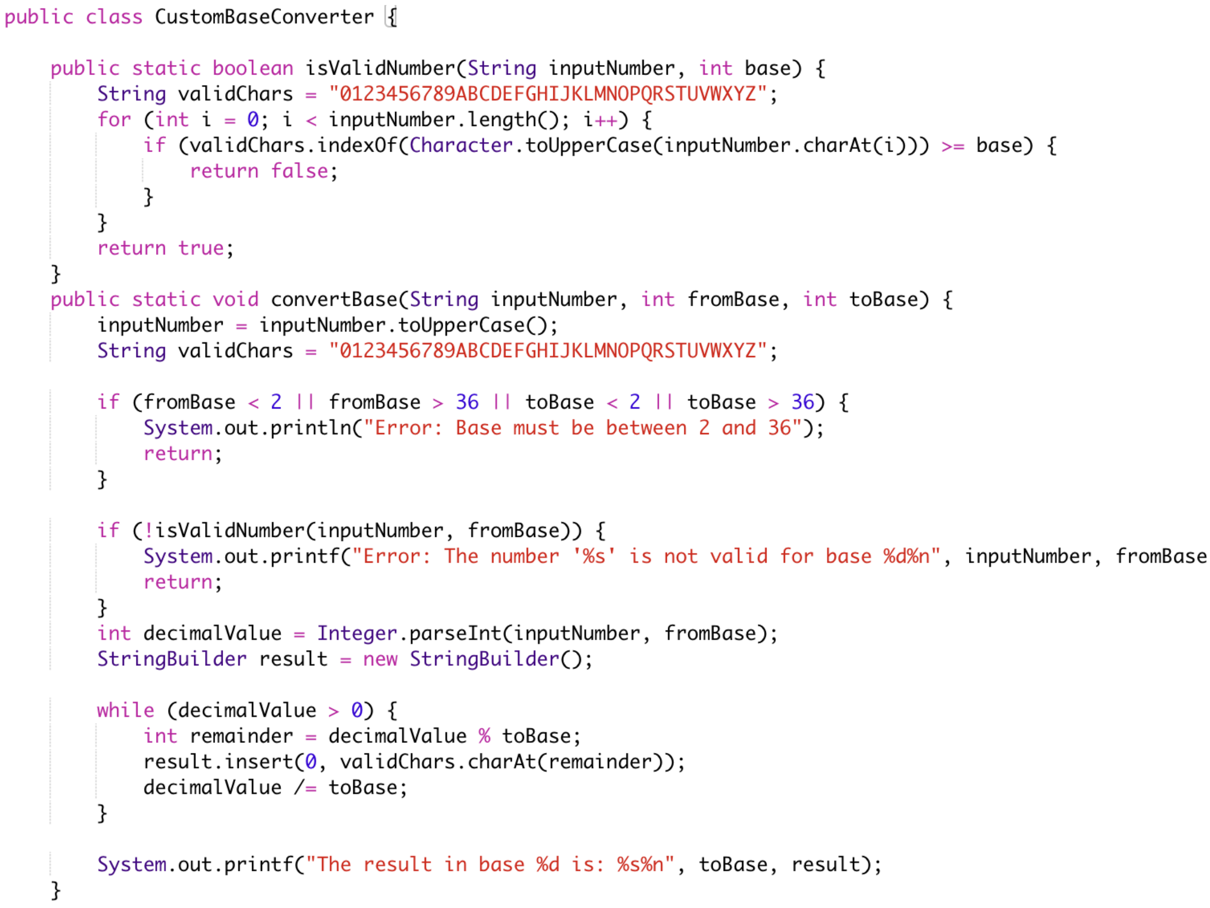
```java
        String binaryString = Integer.toBinaryString(bits);
        while (binaryString.length() < 14) {
            binaryString = "0" + binaryString;
        }

        return binaryString;
    }

    public static void saveToHardwareMemory(String binaryRepresentation) {
        // Implement this method to save the binary representation to hardware memory
        // You can write the necessary code here to interact with hardware memory
        // For the sake of this example, we'll print the binaryRepresentation to the console
        System.out.println("Saving to hardware memory: " + binaryRepresentation);
    }

    public static void main(String[] args) {
        float givenDecimalNumber = -12.345f; // User input
        String binaryRepresentation = convertToFloatingValue(givenDecimalNumber);
        System.out.println("14-bit binary floating value: " + binaryRepresentation);
        saveToHardwareMemory(binaryRepresentation);
    }
}
```



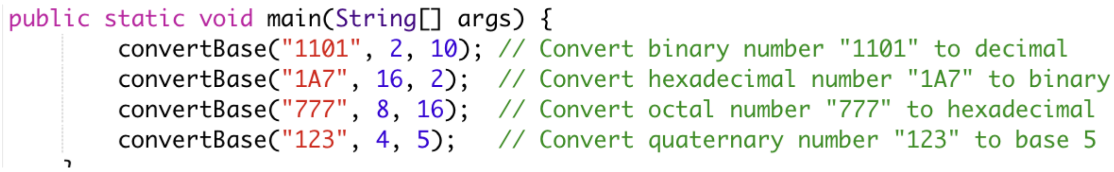
```
14-bit binary floating value: 10010111000101100
Saving to hardware memory: 10010111000101100
```

Question 2

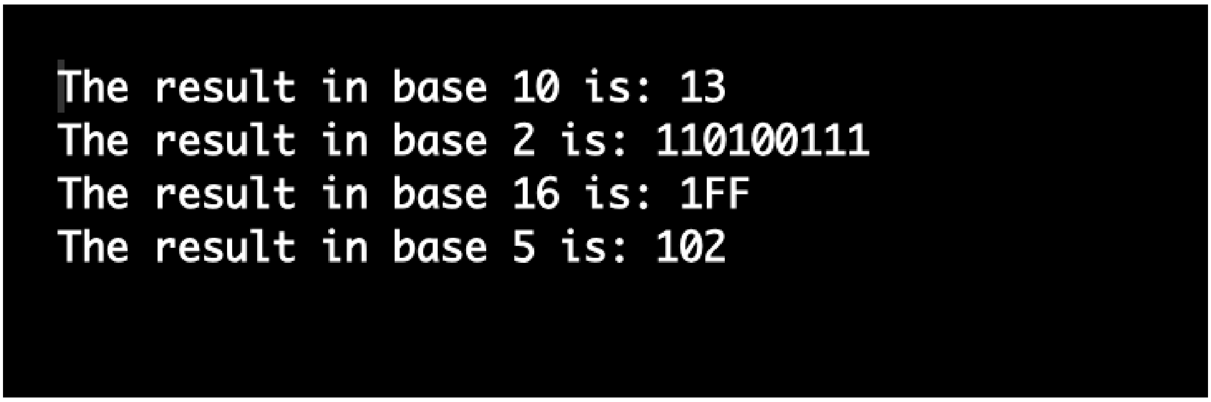
Question 1



```java
public class CustomBaseConverter {

    public static boolean isValidNumber(String inputNumber, int base) {
        String validChars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        for (int i = 0; i < inputNumber.length(); i++) {
            if (validChars.indexOf(Character.toUpperCase(inputNumber.charAt(i))) >= base) {
                return false;
            }
        }
        return true;
    }
    public static void convertBase(String inputNumber, int fromBase, int toBase) {
        inputNumber = inputNumber.toUpperCase();
        String validChars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        if (fromBase < 2 || fromBase > 36 || toBase < 2 || toBase > 36) {
            System.out.println("Error: Base must be between 2 and 36");
            return;
        }

        if (!isValidNumber(inputNumber, fromBase)) {
            System.out.printf("Error: The number '%s' is not valid for base %d%n", inputNumber, fromBase
            return;
        }
        int decimalValue = Integer.parseInt(inputNumber, fromBase);
        StringBuilder result = new StringBuilder();

        while (decimalValue > 0) {
            int remainder = decimalValue % toBase;
            result.insert(0, validChars.charAt(remainder));
            decimalValue /= toBase;
        }

        System.out.printf("The result in base %d is: %s%n", toBase, result);
    }
```



```java
public static void main(String[] args) {
        convertBase("1101", 2, 10); // Convert binary number "1101" to decimal
        convertBase("1A7", 16, 2);  // Convert hexadecimal number "1A7" to binary
        convertBase("777", 8, 16);  // Convert octal number "777" to hexadecimal
        convertBase("123", 4, 5);   // Convert quaternary number "123" to base 5
    }
```



```
The result in base 10 is: 13
The result in base 2 is: 110100111
The result in base 16 is: 1FF
The result in base 5 is: 102
```