

2(a)

main.cpp

Run

```
3 #include <bitset>
4 #include <iomanip>
5
6 int main() {
7     int decimalNumbers[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
        , 13, 14, 15, 16, 17, 18, 31, 100, 255, 256};
8     int size = sizeof(decimalNumbers) / sizeof(decimalNumbers[0]);
9
10    std::cout << "Decimal\t\tBinary\t\tHexadecimal" << std::endl;
11    std::cout << "-----\t\t-----\t\t-----" << std::endl;
12
13    for (int i = 0; i < size; ++i) {
14        int decimal = decimalNumbers[i];
15        std::bitset<8> binary(decimal);
16        std::cout << std::setw(2) << decimal << "\t\t" << binary <<
            "\t\t"
17        << "0x" << std::hex << std::uppercase << decimal
            << std::dec << std::endl;
18    }
19
20    return 0;
```

/tmp/IzyB3ZIzBK.o

Decimal	Binary	Hexadecimal
-----	-----	-----
0	00000000	0x0
1	00000001	0x1
2	00000010	0x2
3	00000011	0x3
4	00000100	0x4
5	00000101	0x5
6	00000110	0x6
7	00000111	0x7
8	00001000	0x8
9	00001001	0x9
10	00001010	0xA
11	00001011	0xB
12	00001100	0xC
13	00001101	0xD
14	00001110	0xE
15	00001111	0xF
16	00010000	0x10
17	00010001	0x11

2(b)

```
1 import java.util.Random;
2 public class FloatingPointConversion {
3     public static void main(String[] args) {
4         Random random = new Random();
5
6         System.out.println("S/No.\tDecimal Number\t\tBinary Number\t\tRemarks\n");
7
8         // Generate and process 50 random floating-point numbers
9         for (int i = 1; i <= 50; i++) {
10             // Generate a random decimal number between 0 and 1000 with at most three decimal points
11             double decimalNumber = random.nextDouble() * 1000.0;
12             decimalNumber = Math.round(decimalNumber * 1000.0) / 1000.0;
13             // Round to three decimal points
14
15             // Convert the decimal number to binary notation
16             String binaryNumber = convertToBinary(decimalNumber);
17
18             // Determine the remarks based on the binary number's length
19             String remarks = getRemarks(binaryNumber);
20
21             // Print the result in a formatted manner
22             System.out.printf("%d.\t%.3f\t\t%s\t\t%s\n", i, decimalNumber,
23                             binaryNumber, remarks);
24         }
25     }
26
27     // Convert a decimal number to binary notation
28     public static String convertToBinary(double decimal) {
29         long intPart = (long) decimal;
30         double fracPart = decimal - intPart;
31
32         // Convert the integer part to binary
33         StringBuilder binaryIntPart = new
34             StringBuilder(Long.toBinaryString(intPart));
35
36         // Convert the fractional part to binary
37         StringBuilder binaryFracPart = new
38             StringBuilder();
39         int count = 0;
40         while (fracPart > 0 && count < 32) {
41             fracPart *= 2;
42             if (fracPart >= 1) {
43                 binaryFracPart.append("1");
44                 fracPart -= 1;
45             } else {
46                 binaryFracPart.append("0");
47             }
48             count++;
49         }
50         return binaryIntPart.toString() + binaryFracPart.toString();
51     }
52
53     // Determine remarks based on binary number length
54     public static String getRemarks(String binaryNumber) {
55         int length = binaryNumber.length();
56         if (length >= 32) {
57             return "Exceeds 32 bits";
58         } else if (length >= 16) {
59             return "Within 16 to 31 bits";
60         } else {
61             return "Within 1 to 15 bits";
62         }
63     }
64 }
```