1. Write a program to perform number conversions using Java.

```
import java.util.Scanner;

public class NumberSystemConversion {

  /* ----------------------------------------
    Decimal to Binary
  ---------------------------------------- */
  static String decimalToBinary(int decimal) {
    if (decimal == 0)
      return "0";

    StringBuilder binary = new StringBuilder();

    while (decimal > 0) {
      binary.append(decimal % 2);
      decimal /= 2;
    }

    return binary.reverse().toString();
  }

  /* ----------------------------------------
    Binary to Decimal
  ---------------------------------------- */
  static int binaryToDecimal(String binary) {
    int decimal = 0;
```

```
    for (int i = 0; i < binary.length(); i++) {
        decimal = decimal * 2 + (binary.charAt(i) - '0');
    }

    return decimal;
}

/* ----------------------------------------
   Decimal to Octal
---------------------------------------- */
static String decimalToOctal(int decimal) {
    return Integer.toOctalString(decimal);
}

/* ----------------------------------------
   Octal to Decimal
---------------------------------------- */
static int octalToDecimal(String octal) {
    int decimal = 0;

    for (int i = 0; i < octal.length(); i++) {
        decimal = decimal * 8 + (octal.charAt(i) - '0');
    }

    return decimal;
}
```

```
/* ----------------------------------------
  Hexadecimal to Binary
---------------------------------------- */
static String hexadecimalToBinary(String hex) {
   int decimal = Integer.parseInt(hex, 16);
   return decimalToBinary(decimal);
}

/* ----------------------------------------
  Binary to Hexadecimal
---------------------------------------- */
static String binaryToHexadecimal(String binary) {
   int decimal = binaryToDecimal(binary);
   return Integer.toHexString(decimal).toUpperCase();
}

/* ----------------------------------------
  Main Method
---------------------------------------- */
public static void main(String[] args) {

   Scanner sc = new Scanner(System.in);
   int choice;

   while (true) {
      System.out.println("\nMenu:");
      System.out.println("1. Decimal to Binary");
      System.out.println("2. Binary to Decimal");
```

```
System.out.println("3. Decimal to Octal");
System.out.println("4. Octal to Decimal");
System.out.println("5. Hexadecimal to Binary");
System.out.println("6. Binary to Hexadecimal");
System.out.println("7. Exit");

System.out.print("Enter your choice: ");
choice = sc.nextInt();

if (choice == 7) {
   System.out.println("Goodbye!");
   break;
}

switch (choice) {

   case 1:
      System.out.print("Enter a decimal number: ");
      int dec1 = sc.nextInt();
      System.out.println("Decimal to Binary: " + decimalToBinary(dec1));
      break;

   case 2:
      System.out.print("Enter a binary number: ");
      String bin1 = sc.next();
      System.out.println("Binary to Decimal: " + binaryToDecimal(bin1));
      break;
```

```
        case 3:
            System.out.print("Enter a decimal number: ");
            int dec2 = sc.nextInt();
            System.out.println("Decimal to Octal: " + decimalToOctal(dec2));
            break;

        case 4:
            System.out.print("Enter an octal number: ");
            String oct = sc.next();
            System.out.println("Octal to Decimal: " + octalToDecimal(oct));
            break;

        case 5:
            System.out.print("Enter a hexadecimal number: ");
            String hex = sc.next();
            System.out.println("Hexadecimal to Binary: " + hexadecimalToBinary(hex));
            break;

        case 6:
            System.out.print("Enter a binary number: ");
            String bin2 = sc.next();
            System.out.println("Binary to Hexadecimal: " + binaryToHexadecimal(bin2));
            break;

        default:
            System.out.println("Invalid choice!");
    }
}
```

```
        sc.close();
    }
}
```

OUTPUT:

Menu:

1. Decimal to Binary

2. Binary to Decimal

3. Decimal to Octal

4. Octal to Decimal

5. Hexadecimal to Binary

6. Binary to Hexadecimal

7. Exit

Enter your choice: 1

Enter a decimal number: 25

Decimal to Binary: 11001

2. **Write a Java program to perform arithmetic operations**

```
import java.util.Scanner;

public class BinaryOperations {

  /* performs binary addition for the given values */
  static int binaryAddition(int n1, int n2) {
    int carry;
    while (n2 != 0) {
      // calculating the carry and left shift
      carry = (n1 & n2) << 1;
      // calculating the sum
      n1 = n1 ^ n2;
      n2 = carry;
    }
    return n1;
  }

  /* performs binary subtraction for the given values */
  static int binarySubtraction(int n1, int n2) {
    int carry;
    // finding two's complement of n2
    n2 = binaryAddition(~n2, 1);

    while (n2 != 0) {
      carry = (n1 & n2) << 1;
      n1 = n1 ^ n2;
      n2 = carry;
```

```
    }
    return n1;
}

/* performs binary multiplication for the given values */
static int binaryMultiplication(int n1, int n2) {
    int res = 0;
    for (int i = 0; i < n2; i++) {
        res = binaryAddition(res, n1);
    }
    return res;
}

/* performs binary division for the given values */
static int binaryDivision(int n1, int n2) {
    int res, count = 0, twosComplement;

    if (n1 < n2) {
        System.out.println("Division of " + n1 + " and " + n2 + " is 0");
        return 0;
    }

    res = n1;
    twosComplement = binaryAddition(~n2, 1);

    while (res > 0) {
        res = binaryAddition(res, twosComplement);
        if (res <= 0) {
```

```
        if (res == 0)
            count = binaryAddition(count, 1);
        break;
      } else {
        count = binaryAddition(count, 1);
      }
    }
    return count;
}

/* Main Method */
public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    int n1, n2, res, ch;

    while (true) {
      System.out.println("1. Binary Addition");
      System.out.println("2. Binary Subtraction");
      System.out.println("3. Binary Multiplication");
      System.out.println("4. Binary Division");
      System.out.println("5. Exit");
      System.out.print("Enter your choice: ");
      ch = sc.nextInt();

      if (ch == 5) {
        System.out.println("Exiting...");
        break;
```

```
}

System.out.print("Enter the value for n1 and n2: ");
n1 = sc.nextInt();
n2 = sc.nextInt();

switch (ch) {
   case 1:
      res = binaryAddition(n1, n2);
      break;

   case 2:
      res = binarySubtraction(n1, n2);
      break;

   case 3:
      res = binaryMultiplication(n1, n2);
      break;

   case 4:
      res = binaryDivision(n1, n2);
      break;

   default:
      System.out.println("Wrong option!!");
      continue;
}
```

```
            System.out.println("Result: " + res + "\n");
        }


        sc.close();
    }
}
```

OUTPUT:

1. Binary Addition

2. Binary Subtraction

3. Binary Multiplication

4. Binary Division

5. Exit

Enter your choice: 1

Enter the value for n1 and n2: 5 3

Result: 8

3. Implement an absolute loader in C.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

void main()
{
   char input[10];
   int start, length, address;
   FILE *fp1, *fp2;

   clrscr();

   /* Open input and output files */
   fp1 = fopen("input.dat", "r");
   fp2 = fopen("output.dat", "w");

   /* Read first record */
   fscanf(fp1, "%s", input);

   /* Continue until End record (E) */
   while (strcmp(input, "E") != 0)
   {
      /* Header record */
      if (strcmp(input, "H") == 0)
      {
         fscanf(fp1, "%d", &start);
         fscanf(fp1, "%d", &length);
```

```
        fscanf(fp1, "%s", input);
    }

    /* Text record with address */
    else if (strcmp(input, "T") == 0)
    {
        fscanf(fp1, "%d", &address);
        fscanf(fp1, "%s", input);

        fprintf(fp2, "%d\t%c%c\n", address,   input[0], input[1]);
        fprintf(fp2, "%d\t%c%c\n", address+1, input[2], input[3]);
        fprintf(fp2, "%d\t%c%c\n", address+2, input[4], input[5]);

        address += 3;
        fscanf(fp1, "%s", input);
    }

    /* Remaining object code */
    else
    {
        fprintf(fp2, "%d\t%c%c\n", address,   input[0], input[1]);
        fprintf(fp2, "%d\t%c%c\n", address+1, input[2], input[3]);
        fprintf(fp2, "%d\t%c%c\n", address+2, input[4], input[5]);

        address += 3;
        fscanf(fp1, "%s", input);
    }
}
```

```
    /* Close files */
    fclose(fp1);
    fclose(fp2);

    printf("FINISHED");
    getch();
}
```

INPUT FILE:

INPUT.DAT H 1000 232

T 1000 142033 483039 102036

T 2000 298300 230000 282030 302015 E

OUTPUT FILE:

OUTPUT.DAT 1000 14

1001 20

1002 33

1003 48

1004 30

1005 39

1006 10

1007 20

1008 36

2000 29

2001 83

2002 00

2003 23

2004 00

2005 00

2006 28

2007 20

2008 30

2009 30

2010 20

2011

4. Implement a relocating loader in C.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

void main()
{
   char add[6], length[10], input[10];
   char binary[12], bitmask[12], relocbit;

   int start, inp, len, i;
   int address, opcode, addr, actualadd;

   FILE *fp1, *fp2;

   clrscr();

   /* Get actual starting address */
   printf("Enter the actual starting address : ");
   scanf("%d", &start);

   /* Open input and output files */
   fp1 = fopen("relinput.dat", "r");
   fp2 = fopen("reloutput.dat", "w");

   /* Read first record */
   fscanf(fp1, "%s", input);

   /* Continue until End record */
   while (strcmp(input, "E") != 0)
   {
      /* Header record */
      if (strcmp(input, "H") == 0)
      {
         fscanf(fp1, "%s", add);
         fscanf(fp1, "%s", length);
         fscanf(fp1, "%s", input);
      }

      /* Text record */
      if (strcmp(input, "T") == 0)
```

```
        {
            fscanf(fp1, "%d", &address);
            fscanf(fp1, "%s", bitmask);

            /* Adjust starting address */
            address += start;

            len = strlen(bitmask);

            /* Process each relocation bit */
            for (i = 0; i < len; i++)
            {
                fscanf(fp1, "%d", &opcode);
                fscanf(fp1, "%d", &addr);

                relocbit = bitmask[i];

                if (relocbit == '0')
                    actualadd = addr;
                else
                    actualadd = addr + start;

                fprintf(fp2, "%d\t%d%d\n", address, opcode, actualadd);
                address += 3;
            }

            fscanf(fp1, "%s", input);
        }
    }

    /* Close files */
    fclose(fp1);
    fclose(fp2);

    printf("FINISHED");
    getch();
}
```

OUTPUT:

Enter the actual starting address :4000

RELOUTPUT.DAT 4000 144033

4003 484039

4006 901776

4009 921765

4012 574765

5011 234838

5014 434979

5017 894060

5020 664849

5023 991477

5. Write a program to perform register operations in Java.

```
import java.util.Scanner;

public class MicroOperations {

    // Simulate 8-bit registers
    static int R1 = 0x0F; // 00001111
    static int R2 = 0xF0; // 11110000

    static void showRegisters() {
        System.out.printf("R1: 0x%02X\tR2: 0x%02X\n", R1 & 0xFF, R2 & 0xFF);
    }

    static void transfer() {
        R1 = R2 & 0xFF;
        System.out.println("Performed: R1 ← R2");
        showRegisters();
    }

    static void add() {
        R1 = (R1 + R2) & 0xFF;
        System.out.println("Performed: R1 ← R1 + R2");
        showRegisters();
    }

    static void bitwiseAnd() {
        R1 = (R1 & R2) & 0xFF;
        System.out.println("Performed: R1 ← R1 AND R2");
        showRegisters();
```

```
}

static void bitwiseOr() {
    R1 = (R1 | R2) & 0xFF;
    System.out.println("Performed: R1 ← R1 OR R2");
    showRegisters();
}

static void leftShift() {
    R1 = (R1 << 1) & 0xFF;
    System.out.println("Performed: R1 ← R1 << 1 (Left Shift)");
    showRegisters();
}

static void rightShift() {
    R1 = (R1 >> 1) & 0xFF;
    System.out.println("Performed: R1 ← R1 >> 1 (Right Shift)");
    showRegisters();
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    int choice;

    while (true) {
        System.out.println("\n--- Micro-Operations Menu ---");
        System.out.println("1. Register Transfer (R1 ← R2)");
        System.out.println("2. Addition (R1 ← R1 + R2)");
```

```
System.out.println("3. Bitwise AND (R1 ← R1 AND R2)");
System.out.println("4. Bitwise OR (R1 ← R1 OR R2)");
System.out.println("5. Left Shift (R1 << 1)");
System.out.println("6. Right Shift (R1 >> 1)");
System.out.println("7. Reset Registers");
System.out.println("8. Exit");
System.out.print("Choose an operation: ");

choice = sc.nextInt();

switch (choice) {
    case 1: transfer(); break;
    case 2: add(); break;
    case 3: bitwiseAnd(); break;
    case 4: bitwiseOr(); break;
    case 5: leftShift(); break;
    case 6: rightShift(); break;
    case 7:
        R1 = 0x0F;
        R2 = 0xF0;
        System.out.println("Registers reset to default values.");
        showRegisters();
        break;
    case 8:
        System.out.println("Exiting program.");
        sc.close();
        return;
    default:
        System.out.println("Invalid choice. Try again.");
```

```
            }
        }
    }
}
```

OUTPUT:

--- Micro-Operations Menu ---

1. Register Transfer (R1 ← R2)

2. Addition (R1 ← R1 + R2)

3. Bitwise AND (R1 ← R1 AND R2)

4. Bitwise OR (R1 ← R1 OR R2)

5. Left Shift (R1 << 1)

6. Right Shift (R1 >> 1)

7. Reset Registers

8. Exit

Choose an operation: **2**

Performed: R1 ← R1 + R2

R1: 0xFF R2: 0xF0

--- Micro-Operations Menu ---

1. Register Transfer (R1 ← R2)

2. Addition (R1 ← R1 + R2)

3. Bitwise AND (R1 ← R1 AND R2)

4. Bitwise OR (R1 ← R1 OR R2)

5. Left Shift (R1 << 1)

6. Right Shift (R1 >> 1)

7. Reset Registers

8. Exit

Choose an operation: **8**

Exiting program.

6. Program to print ASCII for characters in Java.

```
public class ASCII {
   public static void main(String[] args) {
      for (int i = 0; i < 128; i++) {
         if (i >= 32 && i <= 126)
            System.out.println((char) i + " = " + i);
         else
            System.out.println("Non printable = " + i);
      }
   }
}
```

OUTPUT:

Non printable = 0

Non printable = 1

Non printable = 2

Non printable = 3

Non printable = 4

Non printable = 5

Non printable = 6

Non printable = 7

Non printable = 8

Non printable = 9

Non printable = 10

Non printable = 11

Non printable = 12

Non printable = 13

Non printable = 14

Non printable = 15

Non printable = 16

Non printable = 17

Non printable = 18

Non printable = 19

Non printable = 20

Non printable = 21

Non printable = 22

Non printable = 23

Non printable = 24

Non printable = 25

Non printable = 26

Non printable = 27

Non printable = 28

Non printable = 29

Non printable = 30

Non printable = 31

 = 32

! = 33

" = 34

= 35

$ = 36

% = 37

& = 38

' = 39

(= 40

) = 41

* = 42

+ = 43

, = 44

- = 45

. = 46

/ = 47

0 = 48

1 = 49

2 = 50

3 = 51

4 = 52

5 = 53

6 = 54

7 = 55

8 = 56

9 = 57

: = 58

; = 59

< = 60

= = 61

> = 62

? = 63

@ = 64

A = 65

B = 66

C = 67

D = 68

E = 69

F = 70

G = 71

H = 72

I = 73

J = 74

K = 75

L = 76

M = 77

N = 78

O = 79

P = 80

Q = 81

R = 82

S = 83

T = 84

U = 85

V = 86

W = 87

X = 88

Y = 89

Z = 90

[= 91

\ = 92

] = 93

^ = 94

_ = 95

` = 96

a = 97

b = 98

c = 99

d = 100

e = 101

f = 102

g = 103

h = 104

i = 105

j = 106

k = 107

l = 108

m = 109

n = 110

o = 111

p = 112

q = 113

r = 114

s = 115

t = 116

u = 117

v = 118

w = 119

x = 120

y = 121

z = 122

{ = 123

| = 124

} = 125

~ = 126

Non printable = 127

7. Program to implement logic gates in Java.

```
import java.util.Scanner;

public class LogicGates {

    // Logic gate methods
    static int AND(int a, int b) {
        return a & b;
    }

    static int OR(int a, int b) {
        return a | b;
    }

    static int NOT(int a) {
        return (a == 0) ? 1 : 0;
    }

    static int XOR(int a, int b) {
        return a ^ b;
    }

    static int NAND(int a, int b) {
        return (a & b) == 1 ? 0 : 1;
    }

    static int NOR(int a, int b) {
        return (a | b) == 1 ? 0 : 1;
    }
```

```
static int XNOR(int a, int b) {
    return (a ^ b) == 1 ? 0 : 1;
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    int choice, a = 0, b = 0;

    System.out.println("Logic Gates Implementation");
    System.out.println("==========================");
    System.out.println("1. AND");
    System.out.println("2. OR");
    System.out.println("3. NOT");
    System.out.println("4. XOR");
    System.out.println("5. NAND");
    System.out.println("6. NOR");
    System.out.println("7. XNOR");

    System.out.print("\nEnter your choice (1-7): ");
    choice = sc.nextInt();

    if (choice == 3) {
        System.out.print("Enter input (0 or 1): ");
        a = sc.nextInt();
    } else {
        System.out.print("Enter first input (0 or 1): ");
        a = sc.nextInt();
```

```
    System.out.print("Enter second input (0 or 1): ");
    b = sc.nextInt();
}

// Input validation
if ((a != 0 && a != 1) || (choice != 3 && (b != 0 && b != 1))) {
    System.out.println("Invalid input! Inputs must be 0 or 1.");
    sc.close();
    return;
}

switch (choice) {
    case 1:
        System.out.println("AND(" + a + ", " + b + ") = " + AND(a, b));
        break;
    case 2:
        System.out.println("OR(" + a + ", " + b + ") = " + OR(a, b));
        break;
    case 3:
        System.out.println("NOT(" + a + ") = " + NOT(a));
        break;
    case 4:
        System.out.println("XOR(" + a + ", " + b + ") = " + XOR(a, b));
        break;
    case 5:
        System.out.println("NAND(" + a + ", " + b + ") = " + NAND(a, b));
        break;
    case 6:
        System.out.println("NOR(" + a + ", " + b + ") = " + NOR(a, b));
```

```
            break;
        case 7:
            System.out.println("XNOR(" + a + ", " + b + ") = " + XNOR(a, b));
            break;
        default:
            System.out.println("Invalid choice!");
      }

      sc.close();
   }
}
```

OUTPUT:

Logic Gates Implementation

==========

1. AND

2. OR

3. NOT

4. XOR

5. NAND

6. NOR

7. XNOR

Enter your choice (1-7):1

Enter first input (0 or 1): 1

Enter second input (0 or 1): 0

AND(1, 0) = 0

Enter your choice (1-7):2

Enter first input (0 or 1): 1

Enter second input (0 or 1): 0

OR(1, 0) = 1

Enter your choice (1-7):3

Enter input (0 or 1): 1

NOT(1) = 0

Enter your choice (1-7):4

Enter first input (0 or 1): 1

Enter second input (0 or 1): 0

XOR(1, 0) = 1

Enter your choice (1-7):5

Enter first input (0 or 1): 1

Enter second input (0 or 1): 1

NAND(1, 1) = 0

Enter your choice (1-7):6

Enter first input (0 or 1): 1

Enter second input (0 or 1): 1

XNOR(1, 1) = 1

Enter your choice (1-7):6

Invalid input! Inputs must be 0 or 1.

8. **Java Program: Half Adder and Full Adder.**

```
import java.util.Scanner;

public class BinaryAdders {

    // Half Adder
    static void halfAdder(int a, int b) {
        int sum = a ^ b;
        int carry = a & b;

        System.out.println("Half Adder Result:");
        System.out.println("Sum = " + sum);
        System.out.println("Carry = " + carry);
    }

    // Full Adder
    static void fullAdder(int a, int b, int cin) {
        int sum = a ^ b ^ cin;
        int carry = (a & b) | (b & cin) | (a & cin);

        System.out.println("Full Adder Result:");
        System.out.println("Sum = " + sum);
        System.out.println("Carry = " + carry);
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
```

```
int choice;
int a, b, cin;

System.out.println("Binary Adder Menu");
System.out.println("==================");
System.out.println("1. Half Adder");
System.out.println("2. Full Adder");

System.out.print("Enter your choice (1 or 2): ");
choice = sc.nextInt();

if (choice == 1) {
    System.out.print("Enter first input (0 or 1): ");
    a = sc.nextInt();
    System.out.print("Enter second input (0 or 1): ");
    b = sc.nextInt();

    if ((a != 0 && a != 1) || (b != 0 && b != 1)) {
        System.out.println("Invalid input! Inputs must be 0 or 1.");
        sc.close();
        return;
    }

    halfAdder(a, b);

} else if (choice == 2) {
    System.out.print("Enter first input (0 or 1): ");
    a = sc.nextInt();
    System.out.print("Enter second input (0 or 1): ");
```

```
            b = sc.nextInt();
            System.out.print("Enter carry-in input (0 or 1): ");
            cin = sc.nextInt();

            if ((a != 0 && a != 1) || (b != 0 && b != 1) || (cin != 0 && cin != 1)) {
                System.out.println("Invalid input! Inputs must be 0 or 1.");
                sc.close();
                return;
            }

            fullAdder(a, b, cin);

        } else {
            System.out.println("Invalid choice! Please select 1 or 2.");
        }

        sc.close();
    }
}
```

OUTPUT:

```
Binary Adder Menu
==================
1. Half Adder
2. Full Adder
Enter your choice (1 or 2): 2
Enter first input (0 or 1): 1
Enter second input (0 or 1): 1
Enter carry-in input (0 or 1): 1
Full Adder Result:
Sum = 1
Carry = 1
```