

```

import java.util.Scanner;

public class CHMOD {
    public static void main(String args[]) {
        // gets input
        Scanner s = new Scanner(System.in);
        System.out.println("Enter input here:");
        String input1 = s.nextLine().substring(3);
        String input2 = s.nextLine().substring(3);
        String input3 = s.nextLine().substring(3);
        String input4 = s.nextLine().substring(3);
        String input5 = s.nextLine().substring(3);

        System.out.println();

        System.out.print("1. ");
        printOutputLine(input1);

        System.out.print("2. ");
        printOutputLine(input2);

        System.out.print("3. ");
        printOutputLine(input3);

        System.out.print("4. ");
        printOutputLine(input4);

        System.out.print("5. ");
        printOutputLine(input5);
    }

    /**
     * Print the corresponding output line for an input line
     * @param input the input line when running
     */
    public static void printOutputLine(String input) {
        // puts input in array in binary and as permissions
        String[] binary = input.split(" ");
        int[] numbers = new int[4];
        String[] permissions = new String[3];

        numbers[0] = Integer.parseInt(binary[0]);
        for (int i = 1; i < 4; i++) {
            numbers[i] = Integer.parseInt(binary[i]);
        }
    }
}

```

```

        if (numbers[i] < 2)
            binary[i] = "00" + Integer.toBinaryString(numbers[i]);
        else if (numbers[i] < 4)
            binary[i] = "0" + Integer.toBinaryString(numbers[i]);
        else
            binary[i] = Integer.toBinaryString(numbers[i]);
        System.out.print(binary[i] + " ");
        permissions[i - 1] = getPermissions(binary[i]);
    }

    // change permissions based on first digit
    if (numbers[0] == 1)
        permissions[0] = permissions[0].replace('x', 's');
    if (numbers[0] == 2)
        permissions[1] = permissions[1].replace('x', 's');
    if (numbers[0] == 4)
        permissions[2] = permissions[2].replace('x', 't');

    System.out.print("and ");
    for (String permission : permissions) {
        System.out.print(permission + " ");
    }
    System.out.println();
}

/**
 * getPermissions will find the string with permissions from a binary string.
 *
 * @param s the binary string
 * @return the string with permissions
 */
public static String getPermissions(String s) {
    String permissions = "";
    if (s.charAt(0) == '1')
        permissions += "r";
    else
        permissions += "-";

    if (s.charAt(1) == '1')
        permissions += "w";
    else
        permissions += "-";

    if (s.charAt(2) == '1')

```

```
        permissions += "x";
else
    permissions += "-";
return permissions;
    }
}
```

ACSL CHMOD  
Senior Division

**PROBLEM:** CHMOD is a command in the UNIX computer system. It is the command used for giving users permissions to access and change files and directories. There are 3 classes of users. They are the owner, the group and others. The permissions given are: read(r), write(w) and execute(x).

The argument of the CHMOD command is a 3-character octal number (ex. 526). When each digit of that number is converted to binary, the binary digits are paired to represent read, write and execute in that order. 526 would convert to 101 010 110.

The first binary conversion gives the user permissions. The second gives the group permissions. The third gives the others permissions. So here, the owner has read and execute permissions and that is represented by r-x. The group has only write permission given by -w-. The others class has read and write permissions as given by rw-.

Putting all of the above together CHMOD 526 = 101 010 110 = r-x -w- rw-

Further, a fourth octal digit can be added on the left. This digit gives special permissions such as the ability to change passwords, rename files and delete files. The permissions only apply if the class already has the execute permission. A special permission of 0 indicates no special permissions are granted. A special permission of 1 applies only to the owner. A special permission of 2 applies only to the group class. A special permission of 4 applies only to the others class. When the special permissions are applied to either the owner or group classes, the x is changed to an s. When applied to the others class, the x is changed to a t.

**INPUT:** There will be 5 lines of input. Each line will contain 4 octal digits.

**OUTPUT:** For each line of input, print the 9-bit binary conversion and the 9-character permission string. Print a space between each grouping as shown below. Both outputs must be correct to get the point.

## SAMPLE INPUT

1. 0, 5, 2, 6
2. 1, 7, 3, 0
3. 2, 4, 1, 5
4. 4, 2, 3, 4
5. 4, 5, 6, 7

## SAMPLE OUTPUT

1. 101 010 110 and r-x -w- rw-
2. 111 011 000 and rws -wx ---
3. 100 001 101 and r-- --s r-x
4. 010 011 100 and -w- -wx r--
5. 101 110 111 and r-x rw- rwt