**PROBLEM 1**

**Class: Problem1\_RotateRight**

**Description:**

The Problem1\_RotateRight class provides a method to rotate the elements of an array to the right by 2 bits. The class includes a main method to take user input for the array and demonstrate the right rotation.

**Methods:**

1. ***rotateRightBy2Bits():***

***Description:***

This method takes an array of integers and rotates its elements to the right by 2 bits. The rotation is performed in binary representation.

***Parameters:***

* arr (int[]): The array of integers to be rotated.

***Process:***

Converts each element of the array to its binary representation, appends the binary strings, and creates a new string representing the entire array in binary.

Rotates the binary string to the right by 2 bits.

Parses the rotated binary string back to decimal representation and prints the rotated array in both binary and decimal forms.

***Output:***

Displays the original array, its binary representation, the rotated array in binary, and the rotated array in decimal.

1. ***main():***

***Description:***

The main method serves as the entry point for the Problem1\_RotateRight class. It takes user input for the number of elements and the array of integers, and then calls the rotateRightBy2Bits method to demonstrate right rotation by 2 bits.

***Process:***

Prompts the user to enter the number of elements in the array and the array elements.

Creates an array and populates it with user-input integers.

Calls the *rotateRightBy2Bits()* method, passing the array for right rotation.

**Usage Example:**

enter the no. of elements of the array: 5

enter the array elements (integers) with space in between them: 1 2 3 4 5

*The output would be:*

Original Array: 1 2 3 4 5

Original array in Binary: 00000000000000000000000000000001 00000000000000000000000000000010 00000000000000000000000000000011 00000000000000000000000000000100 00000000000000000000000000000101

Rotated array in binary: 01000000000000000000000000000000 11000000000000000000000000000001 01000000000000000000000000000010 11000000000000000000000000000011 01000000000000000000000000000100

Rotated array in Decimal: 1073741824 -2147483647 1073741826 -2147483645 1073741828

*This indicates that the original array [1, 2, 3, 4, 5] is rotated to the right by 2 bits, and the resulting array in decimal representation is [1073741824, -2147483647, 1073741826, -2147483645, 1073741828].*

**PROBLEM 2**

**Class: Problem2\_ConvertToAnyBase**

**Description:**

The Problem2\_ConvertToAnyBase class provides a method for converting a decimal value to any base. The class includes a main method to take user input and demonstrate the conversion.

**Methods:**

1. ***convertToAnyBase():***

***Description:***

This method takes a decimal value and converts it to the specified base. The supported bases range from 2 to 62, covering a variety of numeral systems including binary, octal, decimal, hexadecimal, and custom bases using alphanumeric characters.

***Parameters:***

* decimalValue (int): The decimal value to be converted.
* base (int): The base to which the decimal value should be converted. The base must be between 2 and 62 (inclusive).

***Returns:***

A string representation of the decimal value in the specified base.

If the provided base is outside the valid range (less than 2 or greater than 62), it returns a message indicating the need for a valid base.

***Conversion Process:***

The method uses a character set that includes digits (0-9), uppercase letters (A-Z), and lowercase letters (a-z) to represent values in bases beyond 36.

The conversion is performed by repeatedly dividing the decimal value by the specified base, obtaining remainders, and appending the corresponding characters to the result string until the decimal value becomes 0.

1. ***main():***

***Description:***

Takes user input for the decimal value and the target base, and displays the result of the conversion.

***Process:***

Prompts the user to enter the decimal value to be converted and the base that the decimal value needs to be converted to.

Calls *convertToBase()* method, passing the decimal value and base for conversion.

**Usage Example:**

Enter the decimal value you want to convert: 1052

Enter the base you want to convert to (up to base-62): 16

*The output would be:*

Decimal 1052 in base 16: 41C

*This indicates that the decimal value 1052 is represented as 41C in base 16.*

ANUBHAV JAISWAL  
SEC : 23412G2

REG NO : 2341010007  
ROLL NO: 18

BRANCH : CSE