| ***Introduction to Computer Programming (2024)***  *Minor Project - 1*  *Description and Output* |
| --- |
|  |

| Name - Kunal Routray  Reg number - 2341018202 |
| --- |

**Problem 1:**

Assume the user enters the values 10, 11, 12 and 13.

**Example Execution:**

1. **User Input:**

Enter 4 integer values for the original array:

Value 1: 10

Value 2: 11

Value 3: 12

Value 4: 13

2. **Initialization:**

1. The program initializes an array named **originalArray** with the user-provided elements: **[10, 11, 12, 13]**.

3. **Printing Original Array:**

**Original array:**

**10 11 12 13**

4. **Printing Original Array in Binary:**

**Original array showing 32 bit binary value:**

**00000000000000000000000000001010**

**00000000000000000000000000001011**

**00000000000000000000000000001100**

**00000000000000000000000000001101**

5. **Rotation:**

1. The **rotateRightBy2Bits** method is called to perform the rotation.

2. The last element (13) is temporarily stored in **lastElement**.

3. Each element, starting from the end, is shifted to the right by 2 bits, preserving the bits and handling overflow.

4. The first element is then rotated with the original last element.

6. **Printing Rotated Array in Binary:**

**After rotation array showing 32 bit binary value:**

**01000000000000000000000000000010**

**10000000000000000000000000000010**

**11000000000000000000000000000011**

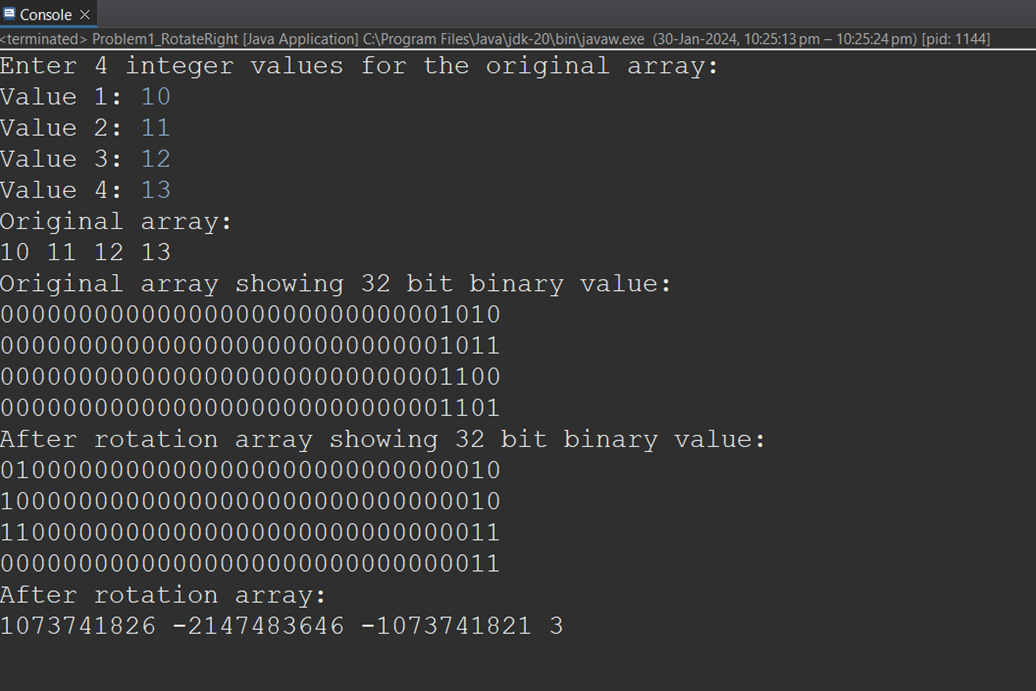
**00000000000000000000000000000011**

7. **Printing Rotated Array:**

**1073741826 -2147483646 -1073741821 3**

**In summary, the program takes user input for an array of 4 values (10, 11, 12, 13), prints it in both integer and binary forms, performs a right rotation by 2 bits, and then prints the rotated array in both integer and binary forms. The example shows how the elements are rotated, considering overflow and the specified rotation pattern based on the user's input.**

**Output:-**

****

**Problem 2:**

Suppose we want to convert the decimal number **42** to base **5**.

1. **Initialization:**

· Decimal number (**n**): 42

· Target base (**b**): 5

· Empty **StringBuilder** named **result**

2. **Loop Execution:**

· In the loop, the remainder of the division (**42 % 5**) is calculated, and the corresponding character from **digitsAndAlphabets** is inserted at the beginning of **result**. The decimal number is then updated to be the result of the integer division (**42 / 5**).

· This process continues until the decimal number becomes zero.

**Iteration 1:**

remainder = 2

result: "2"

decimal number: 8

**Iteration 2:**

remainder = 3

result: "32"

decimal number: 1

**Iteration 3:**

remainder = 1

result: "132"

decimal number: 0

3. **Final Result:**

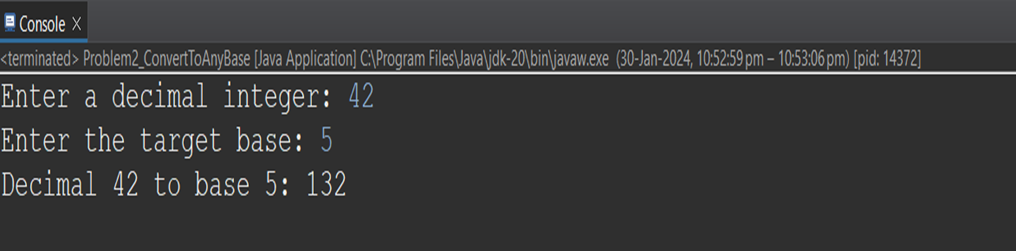
· After the loop, the **StringBuilder** holds the reversed string representation of the number in the target base (**"132"**).

4. **Output:**

· The **StringBuilder** is converted to a string, and the final result is printed.

**Decimal 42 to base 5: "132"**

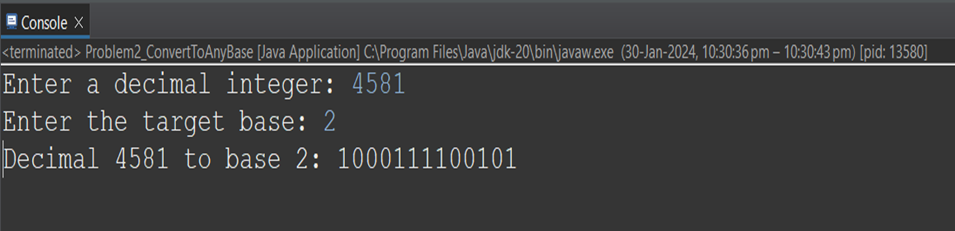
**Output:-**

****

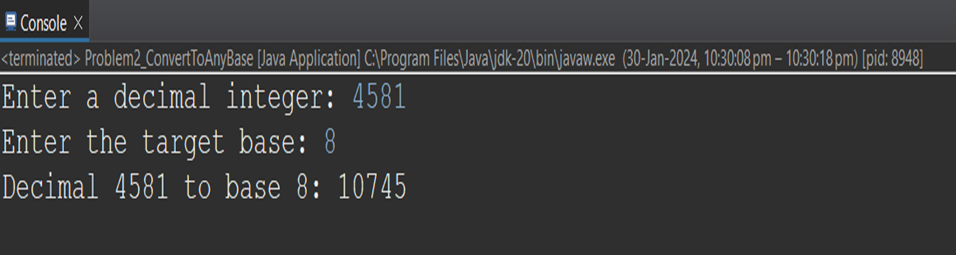
**Taking some more examples**

**Output:-**

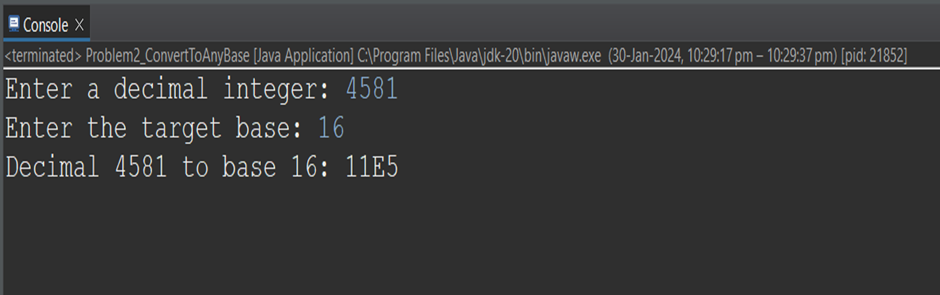
**Decimal to binary:-**

****

**Decimal to octal:-**

****

**Decimal to hexadecimal:-**

****

