

SIKSHA 'O' ANUSANDHAN
DEEMED TO BE UNIVERSITY

Admission Batch: 2023-27

Session 2023-24

MINOR ASSIGNMENT RECORD

Introduction to Computer Programming (CSE-1001)

Submitted By

Name : S K SHAKEEL AKHTAR

Registration Number : 2341001063

Branch : CSE (AIML)

Semester : 1st Semester

Section : 2341-2C3



Department of Computer Science And Engineering

Faculty of Engineering and Technology (ITER)

Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha - 751030

DESCRIPTION:

Problem 1: There are 4 integers in successive memory locations in an array. Write a program to rotate the 4 integers to the right by 2 bits according to the procedure given below. Print the original and resulting array in both integer and binary string form. To print binary string you can use the following library method:

Original Array:

10	11	12	13
----	----	----	----

Original Array 32bit values:

0000....1010	0000....1011	0000....1100	0000....1101
--------------	--------------	--------------	--------------

After rotating 32bit values from right

0100....0010	1000....0010	1100....0011	0000....0011
--------------	--------------	--------------	--------------

After Rotation array:

1073741826	2147483646	-1073741821	3
------------	------------	-------------	---

Description Of Solution:

This Java code defines a class named `Problem1_RotateRight` that implements a right rotation operation on a given integer array.

1. printArray Methods:

- Two overloaded `printArray` methods are defined to print either integer arrays or string arrays. These methods take a message and an array as input, and they print the message followed by the elements of the array.

2. main Method:

- An integer array `n` is declared and initialized with values 10, 11, 12, and 13.
- The `printArray` method is called to print the initial array.

3. rotateRightBy2Bits Method:

- This method takes an integer array as input and performs a right rotation of each element by 2 bits.

- It converts each integer in the array to a 32-bit binary string, ensuring that each binary representation is properly padded with leading zeroes if necessary.
- Two string arrays, `s` and `res`, are initialized to store the binary representations before and after rotation, respectively.
- The method prints the initial binary representations using the `printArray` method.
- It then uses a loop to iterate through each element, calculates the new position after rotation using modular arithmetic, and updates the result array accordingly by concatenating substrings of the binary representation.
- After the rotation, the final binary representations are printed, and the corresponding decimal values are printed as the "final Result".

4. PrefixWithZeroes Method:

- This method takes a binary string as input and ensures that it is represented as a 32-bit binary number by adding leading zeroes if necessary.
- It pads the input string with zeroes on the left until its length becomes 32.

5. Execution in the main Method:

- The `rotateRightBy2Bits` method is called with the initial integer array `n`.
- The initial binary representation of each element is printed.
- The array is then right-rotated by 2 bits, and the rotated binary representations and the final decimal results are printed.

OUTPUT:

```
Input Int array:10 11 12 13

Before Rotation:000000000000000000000000000000001010
                000000000000000000000000000000001011
                000000000000000000000000000000001100
                000000000000000000000000000000001101
After Rotation:010000000000000000000000000000000010
                100000000000000000000000000000000010
                110000000000000000000000000000000011
                0000000000000000000000000000000011 final Result
:1073741826 -2147483646 -1073741821 3
```

DESCRIPTION:

Problem 2: Given a decimal integer (n) and base value (b). Write a Java program to convert n to the corresponding target value n' such that:

$$(n)_{10} = (n')_b$$

[As the target value may contain both alphabets and digits, it is better to use String] Use a method `convertToAnyBase()` with the following header:

```
public static String convertToAnyBase(int n, int b)
```

Description Of Solution:

This Java code defines a class named `Problem2_ConvertToAnyBase` that allows the user to input a decimal number and the target base for conversion. The program then converts the decimal number to the specified base and prints the result.

1. Import Statement:

- The code imports the `Scanner` class from the `java.util` package to take user input.

2. main Method:

- A `Scanner` object (`sc`) is created to take input from the user.
- The user is prompted to enter a decimal number, and the input is stored in the variable `decimalNumber`.
- The user is prompted to enter the target base, and the input is stored in the variable `targetBase`.
- The `convertToAnyBase` method is called with the decimal number and target base as arguments, and the result is stored in the variable `result`.
- The program prints the original decimal number, target base, and the result of conversion.

3. convertToAnyBase Method:

- This method takes two parameters: `n` (the decimal number to be converted) and `b` (the target base).

- It handles the special case where the input decimal number is 0 and returns "0" for this case.
- The method uses a `StringBuilder` named `result` to build the converted representation.
- It enters a loop that continues until the decimal number (`n`) becomes 0.
- In each iteration, it calculates the remainder when dividing `n` by `b` and determines the corresponding digit in the target base.
- The digit is appended to the beginning of the `result` using `insert(0, digit)`.
- The decimal number is updated by dividing it by the target base.
- The loop continues until the decimal number becomes 0.
- Finally, the method returns the converted result as a string.

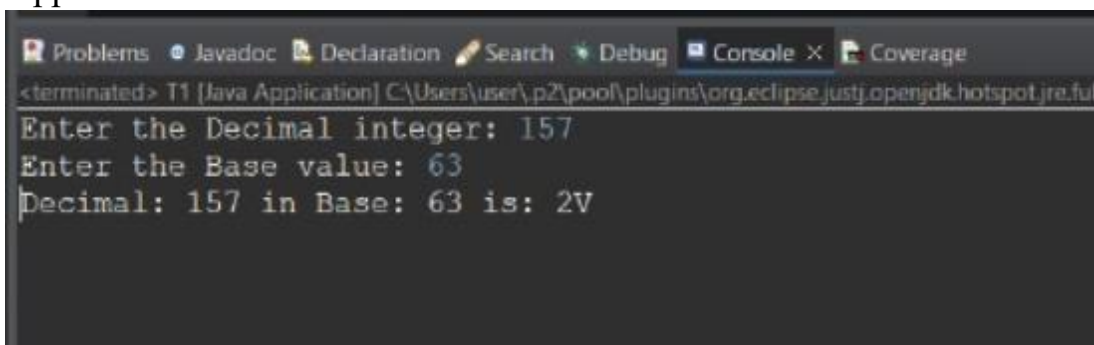
4. Execution in the main Method:

- The user is prompted to enter a decimal number.
- The user is prompted to enter the target base.
- The `convertToAnyBase` method is called with the provided inputs, and the result is printed.

This code provides a flexible solution for converting decimal numbers to any specified base chosen by the user. The conversion includes handling special cases and producing the result in the form of a string.

OUTPUT:

*Suppose we take decimal number =157 and convert it to base 63



```

<terminated> T1 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Enter the Decimal integer: 157
Enter the Base value: 63
Decimal: 157 in Base: 63 is: 2V
  
```

