7. Reverse a Number

Problem: Write a Java program to reverse a given number.

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

Explanation

iNumber: Stores the user-input number.

iSize: Stores the number of digits in iNumber.

iTemp: Temporary variable for manipulating the number.

NewArr[]: Array to store the digits of the reversed number.

**User Input**:

Prompts the user to type a number and stores it in iNumber.

**Determine the Number of Digits**:

Copies iNumber to iTemp.

Makes iTemp positive if it's negative.

Uses a while loop to count the digits by dividing iTemp by 10 until it becomes 0, incrementing iSize with each iteration.

**Initialize the Array**:

Creates a new array NewArr with size iSize.

**Reverse the Number**:

Copies iNumber to iTemp again.

Makes iTemp positive if it's negative.

Uses a while loop to fill NewArr with the digits of iTemp in reverse order by taking the modulus 10 of iTemp, printing the digit, and then dividing iTemp by 10.

Time and space complexity  
0(n)  
  
flowcharts  
 Start

|

v

Type the number (Read input `iNumber`)

|

v

Copy `iNumber` to `iTemp`

|

v

Is `iTemp < 0`? -----> Yes -----> Set `iTemp = -iTemp`

|

No

|

v

While `iTemp > 0`

|

v

Increment `iSize`

|

v

Divide `iTemp` by 10

|

v

End While Loop

|

v

Create `NewArr` of size `iSize`

|

v

Copy `iNumber` to `iTemp`

|

v

Is `iTemp < 0`? -----> Yes -----> Set `iTemp = -iTemp`

|

No

|

v

Initialize `i = 0`

|

v

Print "Reversed number is: "

|

v

Is `iNumber < 0`? -----> Yes -----> Print "-"

|

No

|

v

While `i < iSize`

|

v

Set `NewArr[i] = iTemp % 10`

|

v

Print `NewArr[i]`

|

v

Divide `iTemp` by 10

|

v

Increment `i`

|

v

End While Loop

|

v

Print newline

|

v

End