JAVA LAB Exercises

Date : 27.01.2025

1. Write a program that ignores nonalphanumeric characters in checking whether a string is a palindrome.

   Here are the steps to solve the problem:

   Filter the string by removing the nonalphanumeric characters. This can be done by creating an empty string builder, adding each alphanumeric character in the string
   to a string builder, and returning the string from the string builder. You can use the **isLetterOrDigit(ch)** method in the **Character** class to check whether character **ch** is a letter or a digit.

   Obtain a new string that is the reversal of the filtered string. Compare the reversed string with the filtered string using the **equals** method.

   Sample Runs:

```
Enter a string: ab<c>cb?a Enter
Ignoring nonalphanumeric characters,
is ab<c>cb?a a palindrome? true
```

```
Enter a string: abcc><?cab Enter
Ignoring nonalphanumeric characters,
is abcc><?cab a palindrome? false
```

2. The international standard letter/number mapping found on the telephone is shown below:

Write a program that prompts the user to enter a letter and displays its corresponding number.

```
Enter a letter: A Enter
The corresponding number is 2
```

```
Enter a letter: a Enter
The corresponding number is 2
```

```
Enter a letter: + Enter
+ is an invalid input
```

3. Write a program that prompts the user to enter two characters and displays the major and status represented in the characters. The first character indicates the major and the second is number character 1, 2, 3, 4, which indicates whether a student is a freshman, sophomore, junior, or senior. Suppose the following chracters are used to denote the majors:

   M: Mathematics
   C: Computer Science
   I: Information Technology

   Here is a sample run:

```
Enter two characters: M1 Enter
Mathematics Freshman
```

```
Enter two characters: C3 Enter
Computer Science Junior
```

```
Enter two characters: T3 Enter
Invalid input
```

4. Design a class named StopWatch. The class contains:
   - Private data fields startTime and endTime with getter methods.
   - A no-arg constructor that initializes startTime with the current time.
   - A method named start() that resets the startTime to the current time.
   - A method named stop() that sets the endTime to the current time.
   - A method named getElapsedTime() that returns the elapsed time for the stopwatch in milliseconds.

   Write a test program that measures the execution time of sorting 100,000 numbers using selection sort.