System Requirements Specification Index

For

Java Skills Evaluation

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

USE CASE DESCRIPTION

Use Case 1

Take input of two numbers from users and find out HCF of those numbers.

Input: 10, 15

Output: 5

Description

1. Take two numbers as console input in **main ()** method, from user
2. Pass the numbers to method **getHCFofNumbers()** and write the logic in that method to find HCF of given numbers.
3. Return the HDF from **getHCFofNumbers**() to main() and display it.
4. If any or both the numbers are zero, return hcf as 0
5. Template Code of the same is provided at:

   com.iiht.evaluation.yaksha.hcf.HCFofNumbers

Use Case 2

Take input of a number (N) from the user. You have to perform only two types of operation on number

a) If N is odd, increase it by 1

b) If N is even, divide it by 2

Your task is to find the number of operations it takes to make N equal to 1

Constraints: 1<N<=100000

Input: 5

Output: 5

Input: -4

Output: 0

Description

1. Take console input of number in integer format, in **main ()** method from user.
2. Pass the number to method **getOperationCount()** and write the logic in that method to find operation count.
3. Return the operation count from **getOperationCount**() to main().
4. If the input number is beyond the given limit then method **getOperationCount()** should return 0 .
5. Template Code of the same is provided at:

   com.iiht.evaluation.yaksha.numberoperation.NumberOperation

EXECUTION STEPS

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. This editor Auto Saves the code.
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. Command to run code for use-case 1:

   ```
   mvn compile exec:java -
   Dexec.mainClass="com.iiht.evaluation.yaksha.hcf.HCFofNumbers"
   ```
7. Command to run code for use-case 2:

   ```
   mvn compile exec:java -
   Dexec.mainClass="com.iiht.evaluation.yaksha.numberoperation.NumberOperation"
   ```

8. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

CONSIDERATIONS

1. **Mandatory: Run the following command before the final submission**

   ```
   mvn test
   ```

   (You can run this command multiple times to identify the test case status, and refactor code to make maximum test cases passed before final submission)

2. **Your code will also be evaluated for code quality, naming conventions, readability etc.**
3. **Make sure you do not modify existing class and method names and their signatures, else it would severely affect final result**
4. **Make sure you do not add any new class or methods, else it would severely affect final result**
5. **Make sure you do not modify any test cases, else it would severely affect final result**