|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name: | Laxmareddy A |  | Date: | 22/07/2019 |

|  |
| --- |
| General. |

1. What version of Java (JDK) that you have worked with?

|  |  |
| --- | --- |
| JDK Version | Years of Experience |
| 1.8 | 3 |
| 1.7 | 2 |
| 1.6 | 1 |

2. What is the current JDK version that you are working on?

JDK 1.8

3. What are some of the IDEs and your choice of IDE when developing JAVA applications? Eclipse, STS and Visual Studio Code

4. What are some of the revision source control system that you have used?

|  |  |
| --- | --- |
| Name | Skill Level  (Basic/Intermediate/Advance) |
| SVN | Intermediate |
| Git | Intermediate |
| Mercurial |  |
|  |  |

5. What are some of the Java build tools that you have used?

|  |  |
| --- | --- |
| Name | Skill Level  (Basic/Intermediate/Advance) |
| Apache Ant | Intermediate |
| Apache Maven | Advance |
| Gradle | Basic |
|  |  |

6. What are some of the unit test frameworks/tools that you have used?

|  |  |
| --- | --- |
| Name | Skill Level  (Basic/Intermediate/Advance) |
| JUnit | Intermediate |
| TestNG |  |
| Selenium |  |
| JMeter | Intermediate |

7. What are some of the application frameworks that you have used?

|  |  |  |
| --- | --- | --- |
| Framework | Versions (or Latest Version) | Skill Level  (Basic/Intermediate/Advance) |
| Hibernate | 3.0 | Advance |
| Struts |  | Intermediate |
| Spring | 5.0 | Intermediate |
|  |  |  |
|  |  |  |

8 What kind of J2EE application servers (all the versions) that you have used?

|  |  |  |
| --- | --- | --- |
| Name | Release/Version | Skill Level  (Basic/Intermediate/Advance) |
| Tomcat | 8.0.53 ,8.5.43 | Advance |
| JBoss AS |  |  |
| Glassfish | 4 | Basic |
|  |  |  |
|  |  |  |

9. What kind of database systems (all the versions) that you have worked with?

|  |  |  |
| --- | --- | --- |
| Name | Release/Version | Skill Level  (Basic/Intermediate/Advance) |
| Microsoft SQL Server | 2017 | Advance |
| MySQL | 5.7 | Advance |
| Oracle RDBMS | 10.2(10g) | Intermediate |
| SQLite |  |  |
|  |  |  |
|  |  |  |

10. What kind of continuous integration systems (all the versions) that you have worked with?

|  |  |  |
| --- | --- | --- |
| Name | Release/Version | Skill Level  (Basic/Intermediate/Advance) |
| Jenkins |  | Basic |
| SonarQube |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| Coding Tasks |

*Code and submit the Java source code (one or many files).*

*Each task has two parts: basic and advance.*

*Candidate is expected to complete all the basic sections.*

*Candidate able to complete the advance sections as well will be ranked higher.*

*Grading criteria:*

1. *Correctness*

*• Comprehension - understand the requirement and produce code accordingly.*

*• Bug free - the coding is done without logic bugs.*

*• Performant - able to execute within acceptable timeframe.*

1. *Style*

*• Code formatting - spaces, indentation, braces, new lines.*

*• Naming convention - variables, methods, classes are named meaningfully.*

*• Readability - clear and organized logic flow, easy to read and maintain by others.*

1. *Comprehensiveness*

*• Methods/Functions/Classes/Designs are flexible; cater for reusability and future extension.*

*• Has proper validation and able to handle case scenarios.*

*• Has unit test(s), if possible.*

**Task 1**:

The repayment of a bank housing loan of fixed interest rate is calculated using the fixed rate mortgage formula.

Where:

c is repayment amount

P is the principal amount borrowed

r is the percentage rate per period divided by 100

N is the number of payments

(<https://en.wikipedia.org/wiki/Fixed-rate_mortgage#Pricing>)

For example, a customer securing a loan value of RM 500,000, with fixed interest rate of 4.45% per annum, over a period of 30 years:

The monthly repayment is RM 2,518.59.

**Task 1 (Basic)**:

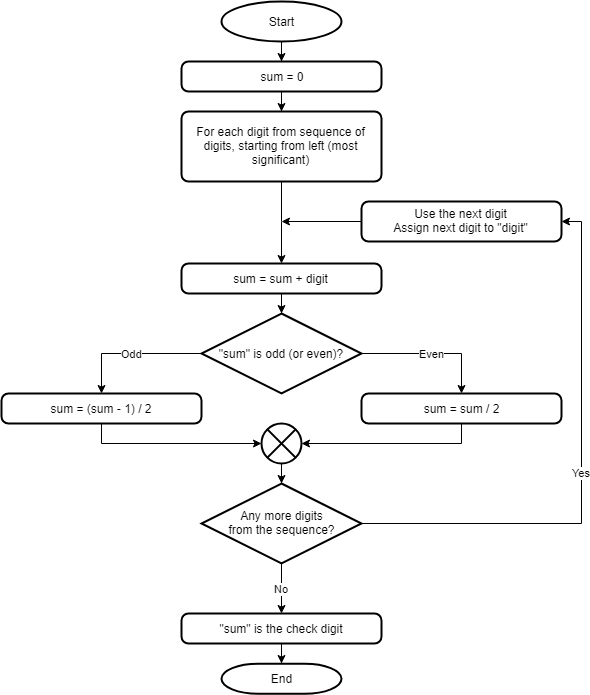
What is the monthly repayment if the customer applying for a loan of RM 400,000, with fixed interest rate of 4.00% per annum, over a period of 30 years?

**Task 1 (Advance)**:

What is the offer of interest rate should the customer look for, if she is able to pay monthly installment of RM 1,500 (or less) per month, when applying for loan of RM 400,000, over a period of 30 years?

**Task 2**:

The check digit algorithm is defined as follow:



For example, given the base value is "2987654321"

The check digit is "1".

The generated value is "29876543211".

**Task 2 (Basic)**:

What is the check digit for the value "98062477123"?

**Task 2 (Advance)**:

Base on the above algorithm, is the calculated check digits (0 to 9) uniformly distributed?

One way to verify: for the number range starting from 1 to 1,000,000, tally the number of each check digits, will the check digits come out (be distributed) equally.

If the check digits are not uniformly distributed, which check digit occurs with the highest frequency?

**Task 3**:

Given the implementation of Linear Congruential Generator to be used as a Pseudo Random Number Generator (PRNG):

class Lcg {

private final long modulus;

private final int multiplier;

private final int increment;

private long seed;

public Lcg(long modulus, int multiplier, int increment, long seed) {

this.modulus = modulus;

this.multiplier = multiplier;

this.increment = increment;

this.seed = seed;

}

// @NotThreadSafe

public /\* synchronized \*/ long next() {

// Y = (a.X + c) mod m

long val = (multiplier \* seed) + increment;

seed = val % modulus;

return seed;

}

}

(<https://en.wikipedia.org/wiki/Linear_congruential_generator>)

(<https://en.wikipedia.org/wiki/Pseudorandom_number_generator>)

**Task 3 (Basic)**:

Given the LCG (modulus = 65536, multiplier = 137, increment = 1, seed = 0), that is producing the sequence of positive integers, test each output number whether it is a prime number or not.

The first few prime numbers would be: 53717, 1789, 23447, 48677, 63611.

What is the 100th prime number (from the sequence using the above LCG)?

(<https://en.wikipedia.org/wiki/List_of_prime_numbers#The_first_1000_prime_numbers>)

**Task 3 (Advance)**:

An encryption logic/function is implemented (as below) using the LCG (modulus = 256, multiplier = 11, increment = 1, seed = 0):

public static String encrypt(String plainText) throws Exception {

Lcg lcg = new Lcg(256, 11, 1, 0);

byte[] bytes = plainText.getBytes("UTF-8");

int len = bytes.length;

byte[] xors = new byte[len];

for (int ix = 0; ix < len; ix += 1) {

xors[ix] = (byte) ((int) bytes[ix] ^ ((int) lcg.next()));

}

return new String(java.util.Base64.getEncoder().encode(xors), "UTF-8");

}

Implement the corresponding decryption logic/function.

public static String decrypt(String base64EncodedValue) throws Exception {

throw new UnsupportedOperationException("TODO: to be implemented");

}