

Home Assignment One

Programming Fundamentals (Java)

Registration # Fa20-Bse-094

Muhammad Ruslan Babar

Submitted to: Sir Azfar Shakeel Khan.

**Question 1:** (Book Question 3.19)

(Business: check ISBN-10) An ISBN-10 (International Standard Book Number) consists of 10 digits: d1d2d3d4d5d6d7d8d9d10. The last digit, d10, is a checksum, which is calculated from the other nine digits using the following formula: (d1 \* 1 + d2 \* 2 + d3 \* 3 + d4 \* 4 + d5 \* 5 +

d6 \* 6 + d7 \* 7 + d8 \* 8 + d9 \* 9) % 11 If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention. Write a program that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros).

**Source Code:**

package russi;

import java.util.\*;

public class ISBN {

public static void main(String[] args) {

//Scanner Object for input

Scanner input = new Scanner(System.in);

System.out.print("Enter the first 9 digits of an ISBN as integer: ");

int userInput = input.nextInt();

int digit\_1 = userInput / 100000000;

int remainingDigits = userInput % 100000000;

int digit\_2 = remainingDigits / 10000000;

remainingDigits %= 10000000;

int digit\_3 = remainingDigits / 1000000;

remainingDigits %= 1000000;

int digit\_4 = remainingDigits / 100000;

remainingDigits %= 100000;

int digit\_5 = remainingDigits / 10000;

remainingDigits %= 10000;

int digit\_6 = remainingDigits / 1000;

remainingDigits %= 1000;

int digit\_7 = remainingDigits / 100;

remainingDigits %= 100;

int digit\_8 = remainingDigits / 10;

remainingDigits %= 10;

int digit\_9 = remainingDigits;

//Final Digit Calculator Formulae

int digit\_10 = (digit\_1 + digit\_2 \* 2 + digit\_3 \* 3 + digit\_4 \* 4 + digit\_5 \* 5

+ digit\_6 \* 6 + digit\_7 \* 7 + digit\_8 \* 8 + digit\_9 \* 9) % 11;

if (digit\_10 == 10)

System.out.printf("The isbn number is %dX",userInput);

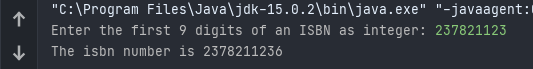
else

System.out.printf("The isbn number is %d%d", userInput , digit\_10);

}

}

**Output shot:**



**Question 2:** (Book Question 3.15)

(Game: lottery) Revise Listing 3.8, Lottery.java, to generate a lottery of a three digit number. The program prompts the user to enter a three-digit number and determine whether the user wins according to the following rules:

* If the user input matches the lottery number in the exact order, the award is $10,000.
* If all digits in the user input match all digits in the lottery number, the award is $3,000.
* If one digit in the user input matches a digit in the lottery number, the award is $1,000.

**Source Code:**

package Russi7kd;

import java.util.\*;

public class Game\_Lottery {

public static void main(String[] args) {

Random random = new Random();

Scanner input = new Scanner(System.in);

//-------------Lottery Number-----------------

int lottery = 100 + random.nextInt(999 - 100);

int randomLottery = lottery;

int part3 = randomLottery %10;

randomLottery /= 10;

int part2 = randomLottery % 10;

randomLottery /= 10;

int part1 = randomLottery;

//---------------User Guess-----------------

System.out.print("Enter three digit guess: ");

int userGuess = input.nextInt();

int userpart3 = userGuess %10;

userGuess /= 10;

int userpart2 = userGuess % 10;

userGuess /= 10;

int userpart1 = userGuess;

//-------------------------------------------

if (userGuess == lottery)

System.out.println("Exact Match\n$10,000 awarded");

else if ((userpart1==part2 || userpart1 ==part3 || userpart1 == part1) && (userpart2==part1 ||

userpart2==part3 || userpart2 == part2) && (userpart3==part1 || userpart3==part2 || userpart3 == part3) )

System.out.println("All matches found\n&3000 awarded");

else if ((userpart1==part2 || userpart1 ==part3 || userpart1 == part1) || (userpart2==part1 || userpart2==part3 || userpart2 == part2) || (userpart3==part1 || userpart3==part2 || userpart3 == part3) )

System.out.println("One Match found\n$1000 awarded");

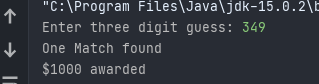
else

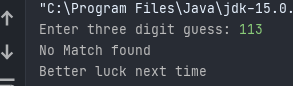
System.out.println("No Match found\nBetter luck next time");

}

}

**Output shot:**





**Question 3:** (Book Question 3.21)

(Science: day of the week) Zeller’s congruence is an algorithm developed by

Christian Zeller to calculate the day of the week. The formula is given in the book, code the formula and test it with different dates to check the result.

**Source Code:**

package russi;

import java.util.\*;

public class ZellersCongruence{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

//Input Year------------------------------

System.out.print("Enter year: (e.g., 2021): ");

int year = input.nextInt();

//Input Month------------------------------

System.out.print("Enter month: 1-12: ");

int m = input.nextInt();

//Special Case for January and February

if (m == 1 || m == 2) {

m = m + 12;

year = year - 1;

}

//Input Month------------------------------

System.out.print("Enter the day of the month: 1-31: ");

int q = input.nextInt();

//Year of Century------------------------------

int k = year % 100;

//Century------------------------------

int j = year / 100;

//Formulae for Day of month Calculations

int h = (q + ( ( 26 \* (m + 1) ) / 10) + k + (k / 4) + (j / 4) + (5 \* j)) % 7;

String day = "";

switch (h) {

case 2: day = "Monday"; break;

case 3: day = "Tuesday"; break;

case 4: day = "Wednesday"; break;

case 5: day = "Thursday"; break;

case 6: day = "Friday"; break;

case 0: day = "Saturday"; break;

case 1: day = "Sunday"; break;

}

//Day displayed------------------------------

System.out.println("Day of the week is " + day);

}

}

**Output Shot:**

