SJSU CS 46A

Final Exam – Fall 2022

Problem 1

Open project 1_ Balance and complete the Java program following the specified steps.

Some constants are provided, and no magic numbers are allowed except in the output format specifiers.

Codecheck link for Problem 1

Problem 2

Open project 2_Member and complete class **`Member`** according to the specified steps. Run the **`main()`** method of class **`MemberTester`** before submitting class **`Member`** to Codecheck.

Codecheck link for Problem 2

Problem 3

Open project 3_College and complete class **`College`** according to the specified steps. Run the **`main()`** method of class **`CollegeTester`** before submitting class **`College`** to Codecheck.

Codecheck link for Problem 3

Problem 4

Open project 4_Department and complete class **`Student`** and class **`Department`** according to the specified steps. Class **`Student`** is basically the same as that in Problem 3, but you will implement interface **`Comparable`**.

Run the **`main()`** method of class **`DepartmentTester`** before submitting both classes **`Student`** and **`Department`** to Codecheck.

Codecheck link for Problem 4

Problem 5

Open project 5_TwoDArray and complete class **`TwoDArrayDouble`** according to the specified steps.

Run the **`main()`** method of class **`TwoDArrayDoubleTester`** before submitting class **`TwoDArrayDouble`** to Codecheck.

Codecheck Link for Problem 5

Problem 6

Many companies send large volumes of mails to their customers. The US Postal Service (USPS) encourages such companies to use a bar encoding for 5-digit zip codes when sending such mails . We stored the bar encodings for all digits in the following 2-D array of ones and zeros.

```
public static int[][] DIGIT_CODE_TABLE = { {1, 1, 0, 0, 0},
                                           {0, 0, 0, 1, 1},
                                           {0, 0, 1, 0, 1},
                                           {0, 0, 1, 1, 0},
                                           {0, 1, 0, 0, 1},
                                           {0, 1, 0, 1, 0},
                                           {0, 1, 1, 0, 0},
                                           {1, 0, 0, 0, 1},
                                           {1, 0, 0, 1, 0},
                                           {1, 0, 1, 0, 0} };
```

Each row of the array represents the bar coding for the row index as a digit, and a one denotes a full bar and a zero denotes a half bar. We will use a `'|'` for a full bar and a `':'` for a half bar. For example, `DIGIT_CODE_TABLE[9]` represents the bar encoding for digit `'9'` and it is `"|:|::"`. Similarly, the bar encoding is `":|:|:"` for `'5'`, `"||:::"` for `'0'`, `":::||"` for `'1'`, and `":|::|"` for digit `'4'`. Then the bar coding for "95014" would be `"|:|:::|:|:||:::::||:|::|"`.

The USPS also adds a checksum digit after the original 5-digit zip code, which will be encoded the same way. The checksum of a 5-digit zip code is calculated as the sum of the five digits, and the checksum digit is the smallest non-negative integer that will make the checksum a multiple of 10 after being added to it. For "95014", the checksum is 19 and the checksum digit is 1. We know the bar encoding for `'1'` is `":::||"`. Notice that if the checksum is a multiple of 10 already, the checksum digit will be 0.

The USPS finally adds a full bar at the start and the end, and the complete bar encoding for "95014" is "`|||:|:::|:|:|||:::::::|||:|::|:::|||`".

Open project 6_ZipcodeEncoding and complete class **`EncodingUtil`** according to the specified steps.

Most likely, you need to convert a digit character `ch` to an int value, and one way to do it is

```
ch - '0'
```

since all digits are listed in the ASCII (and the Unicode) table together starting with '0'.

Run the main() method of class **`EncodingUtilTester`** before submitting class **`EncodingUtil`** to Codecheck.

Codecheck link for Problem 6