Lab 03: Formatted I/O

CSE 4108

Structured Programming I Lab

August 2022

Lab Tasks

1. **It’s a Date:**

Write a program that accepts a date from the user in the format mm/dd/yyyy and then displays it in the form yyyymmdd.

Sample Run:

```
Enter a date (mm/dd/yyyy): 2/17/2021
You entered the data 20210217
```

2. **Inventory Management System:**

Mr. Sakamoto from Convenience Delivery Store (CDS) of Imperial University of Tokyo (IUT) is back. This time, he wants you to create a list of items in his shop. He will provide you item number, unit price and purchase date of the products available in CDS. He wants you to show the items in an organized manner. However, before finalizing the design he wants to see how the list will look like a single item.

You must write a program that formats product information entered by the user. A session with the program should look like this:

```
Enter item number: 583
Enter unit price: 13.5
Enter purchase date (mm/dd/yyyy): 10/24/2019

 Item Price   Date             Unit Purchase

 $ 13.50     10/24/2019                 583
```

The item number and date should be left justified; the unit price should be right justified. Allow dollar amounts up to $9999.99.

3. **Code of ISBN:**

Books are identified by an International Standard Book Number (ISBN). ISBNs assigned after January 1, 2007 contain 13 digits, arranged in five groups, such as **978-0-393-97950-3**. (Older ISBNs use 10 digits). The first group (the GS1 prefix) is currently either **978** or **979**. The group identifier specifies the language or country of origin (for example, **0** and **1** are used in English speaking countries). The *publisher code* identifies the publisher. The item number is assigned by the publisher to identify a specific book. An ISBN ends with a check digit that's used to verify the accuracy of the preceding digits. Write a program that breaks down an ISBN entered by the user.

Sample Run:

```
Enter ISBN: 978-3-16-148410-0

GS1 prefix: 978
Group Identifier: 3
Publisher code: 16
Item number: 148410
Check digit: 0
```

The number of digits in each group may vary; you can't assume that groups have the lengths shown in this example. Test your program with actual ISBN values (usually found on the back cover of a book and on the copyright page).

4. **Phone Number Formats:**

Write a program that prompts the user to enter a telephone number in the form **(xxx)xxx-xxxx** and then displays the number in the form **xxx.xxx.xxxx**.

Sample Run:

```
Enter phone number [(xxx) xxx-xxxx]: (404)817 - 6900
You entered 404.817.6900
```

5. **Numbers Square:**

Write a program that asks the user to enter the numbers from 1 to 16 (in any order) and then displays the numbers in a 4 by 4 arrangement, followed by the sums of the rows, columns, and diagonals.

Sample Run:

```
Enter the numbers from 1 to 16 in any order:
16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1

16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1
```

```
Row sums: 34 34 34 34
Column sums: 34 34 34 34
Diagonal sums: 34 34
```

If the row, column, and diagonal sums are all the same (as they are in this example), the numbers are said to form a magic square. The magic square shown here appears in a 1514 engraving by artist and mathematician Albrecht Dürer. (Note that the middle number in the last row gives the date of the engraving).

6. **Signed Fractions:**

Consider a C program that prompts the user to enter two fractions like the given sample input and then displays their sum. Fractions are customarily denoted as: ***numerator/denominator***

Sample Run:

```
Enter two fractions separated by a plus sign: 5/6+3/4
The sum is: 38/24
```

7. **Digitsum:**

Given a five-digit integer, print the sum of its digits.

Sample Run:

```
Enter a five-digit integer: 10564
The sum is: 16
```