Q1) CheckISBN.py (20) Points)

An ISBN (International Standard Book Number) can consist of 10 or 13 digits. The last digit is called checksum, which is calculated using a certain formula.

Calculations: ISBN-10
The ISBN-10 consists of 10 digits: 0718079183

The last digit is calculated using the first 9 digits as:

$$checksum = (d_1 \times 1 + d_2 \times 2 + d_3 \times 3 + d_4 \times 4 + d_5 \times 5 + d_6 \times 6 + d_7 \times 7 + d_8 \times 8 + d_9 \times 9) \% 11$$

If the checksum is 10, then the 10th digit is denoted as X.

Calculations: ISBN-13
The ISBN-13 consists of 13 digits: 9780718079185

$$checksum = 10 - (d_1 + 3d_2 + d_3 + 3d_4 + d_5 + 3d_6 + d_7 + 3d_8 + d_9 + 3d_{10} + d_{11} + 3d_{12}) \% 10$$

If the checksum is 10, then put 0 as the 13th digit.

Write a program that prompts the user to enter the digits (except the last digit) of an ISBN – it can be either ISBN-10 or ISBN-13. Read the input as a string. User can enter 9 digits or 12 digits. The program should determine the checksum based on the size of the number entered. If the user enters a number that doesn't contain either 9 or 12 digits, display a message and prompt the user to enter 9 or 12 digits.

Sample Output:

```
Enter either 9 or 12 digits as a string (or -1 to quit): 9787312

Invalid ISBN number.
Enter either 9 or 12 digits as a string (or -1 to quit): 978071807918
The ISBN-13 number is: 9780718079185

Enter either 9 or 12 digits as a string (or -1 to quit): 0718079183
The ISBN-10 number is: 0718079183

Enter either 9 or 12 digits as a string (or -1 to quit): 013119402
The ISBN-13 number is: 013119402X

Enter either 9 or 12 digits as a string (or -1 to quit): 978731259722
The ISBN-13 number is: 9787312597220
```

Q2) IndexOfSecondLargestElement.py (10 points)

Write a program that creates a list of 10 integers (between 1 and 20) using a random number generator function. Write a function that returns the index (position) of the second largest element from the list of integers. If there are more than one second largest numbers, return the index of first occurrence of that number.

Eg: Assume that your main function has created the following list of integers generated from the random number function.

[9,18,5,8,3,18,12,20,10,3]

Expected output:

List of integers: 9; 18; 5; 8; 3; 18; 12; 20; 10; 3
Second largest number: 18
Index (position) of second largest number: 1

Q3: Check password: CheckPassword.py (20 points)

Write a program with a main and a sub function called checkPassword that checks whether a string is a valid password. The main function should get the user input as a string.
Define the function as def checkPassword(password). This function returns whether the password is valid or invalid. Use string methods.
Password rules: The password must

- have at least eight characters.
- contain at least two digits.
- must contain at least one uppercase letter
- must contain at least one lowercase letter
- must contain at least one of [!,@,#,$,%,^,&,*,(,)] (non-letter and non-numeric character)
    - Create a list for these special characters

Expected output: Print one of the following based on the value returned from the function.

The password you entered is valid
The password you entered is invalid

Q4: (Average Weekly Sales and Average Sales Per Sales Person): (40 points)

Write two programs CreateSales.py and AverageSales.py. The first program (CreateSales.py) generates a sales file called sales.txt, which contains sales data for 12 weeks by 20 sales people. Sample data is given below. You may use random number function to generate floating point numbers between 0 and 200. Each row represents weekly sales by each sales person for 12 weeks. Data in each column represents sales for a particular week by 20 sales people.

The second program (AverageSales.py) computes average sales for each sales person and for each week.

Sample Data

```
wk1,wk2,wk3,wk4,wk5,wk6,wk7,wk8,wk9,wk10,wk11,wk12
55.8,131.2,12.5,27.6,105.4,181.7,176.3,63.2,179.4,81.6,126.5,122.9,
167.2,26.4,141.7,137.8,114.2,180.3,66.8,46.9,154.7,56.8,21.2,148.6,
...
...
```

Sample Output:

```
Week Total Sales Average Sales
 1       1723        86.15
 2       2064       103.20
 3       2338       116.90
 4       2242       112.10
 .       ...        ....
 .       ...        ....

Sales Person Total Sales     Average Sales
 1           2625              218.75
 2           2462              205.17
 3           4338              361.50
 4           3212              267.67
 .           ...               ....
 .           ...               ....
```

Q5: (Anagrams) Anagram.py (10 Points)

Write a function def isAnagram(s1, s2) that checks whether two words s1 and s2 are anagrams. Two words are anagrams if they contain the same letters. For example, silent and listen; cinema and iceman are anagrams.

Write the main function that prompts the user to enter two strings and displays the output:

Expected output:

```
      Enter the first word: silent
      Enter the second word: listen
      The words “silent” and “listen” are anagrams
```

```
Enter the first word: computer
Enter the second word: processor
The words “computer” and “processor” are not anagrams
```