

## Homework No. 02

Due: 23:59, 07 March, 2023

Max points: 100

### Rules

- **No late homeworks.** A penalty of 10 points is applied for each day.
- **No plagiarism.** Collaboration is encouraged, but copying someone else's work without proper attribution is not admitted and invalidates the submission. A penalty is applied to all parties included.

### Submission procedure

- Each problem solution should be saved in a separate file. The following naming convention should be used: `problem{number}.{extension}`. For example, `problem1.py` or `problem1.pdf`.
- At the start of each file, homework number, student full name and problem number should be mentioned. For example:

```
"""  
Homework 1  
Name: John Doe  
Problem 1  
"""
```

- Solution files should be uploaded to YSU Moodle. Alternatively, you can commit your solutions to a Git repository and provide the repository URL on Moodle.

### Problem 1 [10 points]

Write a Python program that computes and displays the values of the following expressions:

1.

$$\frac{1.2 \times 3.67^4 - 4.3^2 \times 14.89 \times 5.2}{1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9}$$

2.

$$\frac{1^3 + 2^3 + 3^3 + 4^3 + 5^3}{(56 \times 47) \bmod 13 - 4^{56} \bmod 7}$$

### Problem 2 [10 points]

Write a program that allows the users to play the scissor, rock, paper game. The rules are as follows:

- A scissor can cut a paper
- A rock can knock a scissor
- A paper can wrap a rock

The program prompts both players to enter their choices displays a message indicating which player wins or if there is a draw.

The following shows an example program in action.

```
$ python problem2.py
Player 1: rock
Player 2: paper
Player 2 wins.

$ python problem2.py
Player 1: scissor
Player 2: scissor
Player 2 It is a draw.
```

### Problem 3 [10 points]

Body mass index (BMI) is a measure derived from the height and weight of a person. It can be calculated by dividing person's weight in kilograms by the square of their height in meters, i.e.:

$$BMI = \frac{weight}{height^2}$$

The health of a person can be measured using the following interpretation:

Category	BMI
Underweight	$BMI < 18.5$
Normal	$18.5 \leq BMI < 25$
Overweight	$25 \leq BMI < 30$
Obese	$30 \leq BMI$

Write a Python program that asks the user to enter their weight in kilograms and height in meters, and displays their BMI value and category.

The following shows an example program in action.

```
$ python problem3.py
Enter weight in kg: 75
Enter weight in m: 1.8
BMI is 23.15.
BMI category is Normal.
```

#### Problem 4 [10 points]

Write a program that asks the user to enter an integer and outputs **True** if the provided number is a power of 2. Otherwise, it displays **False**.

**Note:** Do not use loops.

The following shows an example program in action.

```
$ python problem4.py
Enter an integer: 8
True

$ python problem4.py
Enter an integer: 9
False
```

#### Problem 5 [10 points]

Write a Python program that asks the user to input a *number* and a value of *n*, which shows a bit position. The program performs the following tasks and displays the results:

1. Set the *n*-th bit of a *number*.
2. Clear the *n*-th bit of a *number*.
3. Toggle the *n*-th bit of a *number*.
4. Check whether the *n*-th bit of a *number* is set or not.

The following shows an example program in action.

```
$ python problem4.py
Enter a number: 42
Enter a value of n: 2
After setting 2th bit: 46
After clearing the 2th bit: 42
After toggling the 2th bit: 46
The 2th bit of 42 is not set
```

### Problem 6 [10 points]

Write a Python program that prompts the user to enter real number values for  $a$ ,  $b$  and  $c$  and solves the following equation:

$$ax^2 + bx + c = 0$$

If the equation has one root or repeated roots, the program should display the single root. If the equation has two roots, both roots should be displayed. If the equation has no real roots, its complex roots should be displayed. Otherwise, the program should display that the equation has no roots.

The following shows an example program in action.

```
$ python problem6.py
Enter a: 1
Enter b: -1
Enter c: -6
The roots are -2 and 3.

$ python problem6.py
Enter a: 0
Enter b: 2.5
Enter c: -4.4
The root is 1.76.
```

### Problem 7 [10 points]

The area of a triangle can be calculated using the following formula:

$$s = \frac{side_1 + side_2 + side_3}{2}$$
$$Area = \sqrt{s(s - side_1)(s - side_2)(s - side_3)}$$

where  $side_1$ ,  $side_2$  and  $side_3$  are the lengths of the sides of the triangle.

Write a Python program that asks the user to enter real number values for  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$  points in the Cartesian coordinate system and computes the area of a triangle that is formed by these vertices. If no such triangle exists, output an appropriate message.

The following shows an example program in action.

```
$ python problem7.py
Enter x_1: 0
Enter y_1: 0
Enter x_2: 2
Enter y_2: 0
Enter x_3: 0
Enter y_3: 1
The area of this triangle is 1.
```

```
$ python problem7.py
Enter x_1: 1
Enter y_1: 1
Enter x_2: 1
Enter y_2: 2
Enter x_3: 1
Enter y_3: 5
No such triangle exists.
```

### Problem 8 [10 points]

To solve a  $2 \times 2$  system of linear equations, we can use Cramer's rule. Given the following system of linear equations,

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned}$$

Cramer's rule provides the solution as follows:

$$x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc}$$

Write a Python program that solves a  $2 \times 2$  system of linear equations by Cramer's rule. It asks the user to enter real number values for  $a, b, c, d, e$ , and  $f$ , and outputs the solution. If no solution exists, an appropriate message should be displayed.

The following shows an example program in action.

```
$ python problem8.py
Enter a: 1
Enter b: 1
Enter c: 1
Enter d: 2
Enter e: 1
Enter f: 2
The solution is (0, 1).

$ python problem8.py
Enter a: 1
Enter b: 2
Enter c: 1
Enter d: 2
Enter e: 1
Enter f: 2
No solution exists.
```

### Problem 9 [10 points]

The following is the formula of temperature conversion from Celsius to Fahrenheit:

$$F = \frac{9}{5} \times C + 32$$

Write a Python program that allows the user to convert temperature from Celsius to Fahrenheit and vice versa. The program should prompt the user to enter whether they want to convert from Celsius to Fahrenheit or vice versa. Then, the user should be able to enter the value of temperature in degrees and receive the result of the required conversion.

The following shows an example program in action.

```
$ python problem9.py
Which conversion to choose (C-F or F-C): C-F
Enter temperature in degrees: 0
The temperature of 0C is 32F.
```

```
$ python problem9.py
Which conversion to choose (C-F or F-C): F-C
Enter temperature in degrees: 0
The temperature of 0F is -17.78.
```



## Problem 10 [10 points]

Zeller's congruence is a mathematical formula created by Christian Zeller for determining the day of the week. The formula is as follows:

$$h = (q + \lfloor \frac{13(m+1)}{5} \rfloor + k + \lfloor \frac{k}{4} \rfloor + \lfloor \frac{j}{4} \rfloor + 5j) \bmod 7$$

where,

- $h$  is the day of the week (0 = Saturday, 1 = Sunday, 2 = Monday, ..., 6 = Friday)
- $q$  is the day of the month (1 - 31)
- $m$  is the month (3 = March, 4 = April, 5 = May, ..., 14 = February). January and February are counted as months 13 and 14 of the previous year.
- $k$  is the year of the century ( $year \bmod 100$ ).
- $j$  is the zero-based century, i.e.  $\lfloor \frac{year}{100} \rfloor$

Write a Python program that asks the user to enter a year, month, and day of the month, and outputs the day of the week.

The following shows an example program in action.

```
$ python problem10.py
Enter year: 2023
Enter month (1 - 12): 2
Enter day of the month (1 - 31): 26
Day of the week is Sunday.
```

## Further reading

1. *Think Python: How to Think Like a Computer Scientist* - Chapter 1, Chapter 2, Chapter 5 (up to **Recursion**)
2. Built-in Types and Operators
3. Bitwise Operators in Python
4. Operators and Expressions in Python
5. Expressions
6. printf-style String Formatting
7. Format String Syntax
8. str.format
9. PEP 498 – Literal String Interpolation
10. Python String Formatting Best Practices
11. PEP 308 – Conditional Expressions