Week 13

Write and run your programs with IDLE editor. Submit finished programs to CodeGrade. Note that some tasks have several steps (A, B, C, …) in CodeGrade.

IMPORTANT: End each input-command string with a newline symbol \n. For example:
`variable = input("Some text:\n")`

Task 1. Write a Python program that takes a filename as a command line argument and checks if the file exists. If the file exists, display its content; otherwise, print an error message. The file **`Ex1_haiku.txt`** can be found in Moodle.

The file that you upload to CodeGrade **must** be `Exercise1.py`.

The program also notifies if the number of command line arguments is not correct, see the example below

Example run 1.

```
Exercises13> python Exercise1.py
Usage: python read_file.py <filename>
```

Example run 2.

```
Exercises13> python Exercise1.py first.txt second.txt
Usage: python read_file.py <filename>
```

Example run 3.

```
Exercises13> python Exercise1.py MyFile.txt
Error: File 'MyFile.txt' does not exist.
```

Example run 4.

```
Exercises13> python Exercise1.py Ex1_haiku.txt
Contents of Ex1_haiku.txt:
Snakes coil in their nest,
Python code, a language blessed,
Simple, yet the best.
```

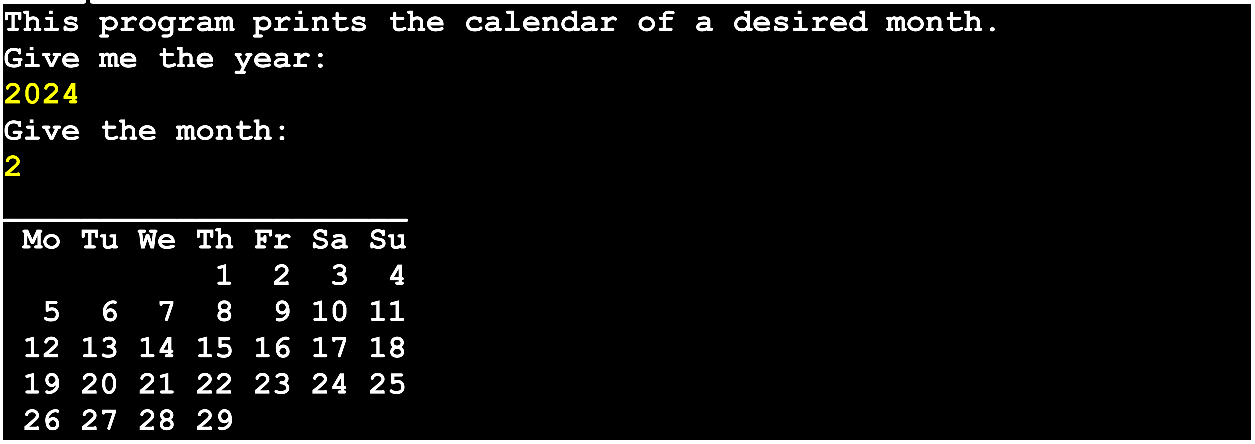
Task 2. Write a program that asks the user for the year and month, and prints the calendar for that month according to the example run below. Implement the solution using the `datetime` module and design a suitable algorithm yourself.

The line containing the weekdays (`Mo Tu We Th Fr Sa Su`) can be printed as one string, because this line will be always the same. Note that first day of the month can be any weekday, so first you have to find out the weekday of the first day of the month. You also need to know the number of days in the month.

The weekday of the first day can be found by using `datetime.date.weekday`. The number of days in a month can be calculated as the difference in dates between the start of the following month and the start of the month in question.

While printing, use 3-character fields. This can be done by using the format-string:
`print(f"{day:3}", end="")`

Example run 1.



```
This program prints the calendar of a desired month.
Give me the year:
2024
Give the month:
2

 Mo Tu We Th Fr Sa Su
           1  2  3  4
  5  6  7  8  9 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25
 26 27 28 29
```

Task 3. *Rock, Paper, Scissors* is a simple hand game usually played between two people, where each player simultaneously forms one of three shapes using their hand. The possible shapes are rock, paper, and scissors. The game has simple rules for determining the winner:

- Rock beats Scissors.
- Scissors beat Paper.
- Paper beats Rock.

Your task is to implement this game. Create a game loop (that's just a regular while loop!) that asks for user input. User should select one of the options from `Rock` / `Paper` / `Scissors` or type `Exit` to stop the program. The game **must** play repeatedly until the player wants to exit.

Initialize the options as a list of strings

```
CHOICES = ["Rock", "Paper", "Scissors"].
```

Use Python's `random`-module to select the computer's answer from the list by

```
CHOICES[random.randint(0,2)]
```

Note: Use `random.seed(42)` at the beginning of the code before random-generator is called for the first time. This way CodeGrade can check the solutions, and your code returns the same answer every time.

Hint: This is probably easiest to implement with multiple `if-elif-else` structures.

Example run 1.

```
Rock, Paper, Scissors (type 'Exit' to quit):
Rock
You won! Rock triumphs Scissors
Rock, Paper, Scissors (type 'Exit' to quit):
Rock
It was a tie!
Rock, Paper, Scissors (type 'Exit' to quit):
Scissors
You lost! Scissors loses to Rock
Rock, Paper, Scissors (type 'Exit' to quit):
eixt
That's not a valid play. Check your spelling!
Rock, Paper, Scissors (type 'Exit' to quit):
Exit
```

Task 4: Your task is to write a **Tic-Tac-Toe** game on 3 x 3 board. These 9 slots are numbered: 1, 2, 3, 4, 5, 6, 7, 8, 9.

You already have a base code `Exercise4_base.py` to start with. You can find this file in Moodle. This file contains also a lot of useful comments.

Computer plays with `O`'s and you play with `X`'s

The basic game loop is already implemented, but it misses certain functions. Fill in the following parts:

`initialize()`: Create a 3x3 numpy array here and fill it with 3s. Number 3 denotes the empty slot.

`choose_starting_player()`: Fill in a coin flip function at the start of game that randomly chooses who goes first, the human player (`1`) or the computer (`0`). The submitted program **must** use `coin_flip = random.randint(0, 1).`

`display_board()`: Fill in the missing part of code that enables printing of the 3 x 3 game board. Just create a list that has 9 entries that are either `X, O` or `' '` (empty slot). Using this list, printing the symbols in the board is easy.

`place_symbol()`: Write code to place the letter of the active player to the proper spot in the 3 x 3 board. For example, if player chooses 5 (the center spot in the grid), the code should place `board[1][2] = 1`, where `0=O, 1=X, 3=' '`.

`player_turn()`: Read the user input according to the example below.

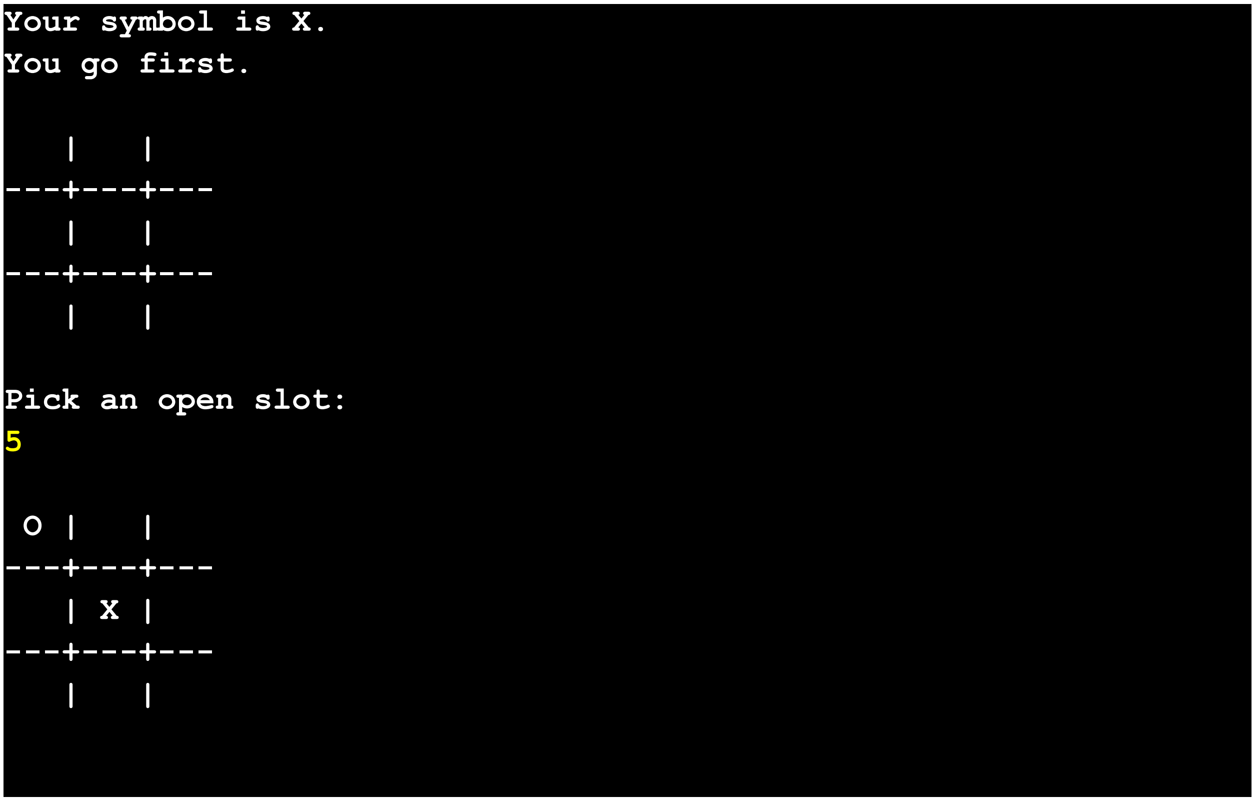
All missing code parts are marked in `py`-file with a comment line starting with `# TODO`:

Note 1. If you want to really play the game, comment out the `random.seed(0)` at the top of the code. That way computer starts to play differently (but it is not very clever). Be sure that when you submit the task to CodeGrade, the seed **must** be set to 0.

Note 2. CodeGrade has `NumPy` installed for this task.

Note 3. This exercise gives **6 points**.

Example run 1 (no given seed).

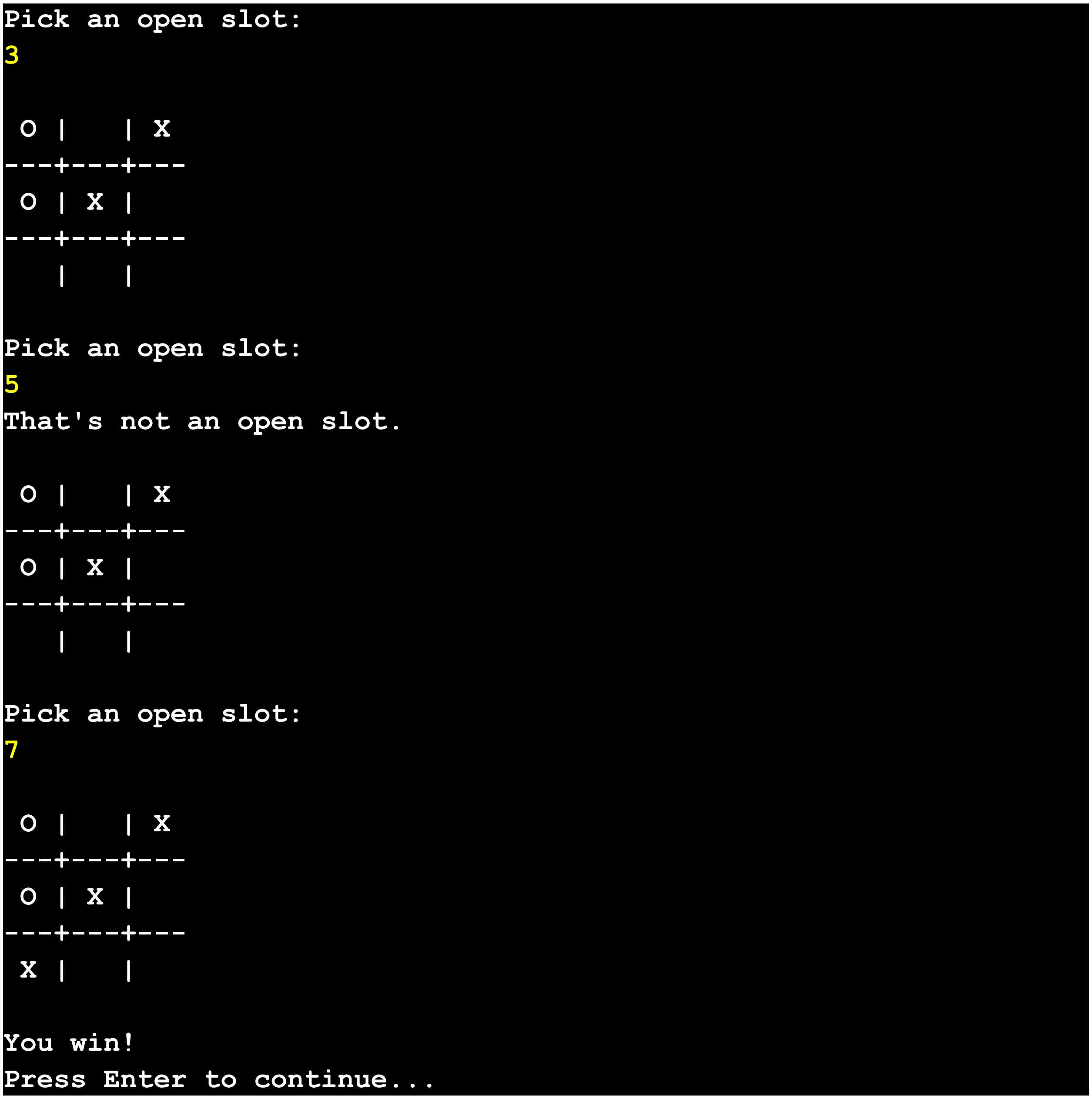


```
Your symbol is X.
You go first.

   |   |
---+---+---
   |   |
---+---+---
   |   |

Pick an open slot:
5

 O |   |
---+---+---
   | X |
---+---+---
   |   |
```



```
Pick an open slot:
3

 O |   | X
---+---+---
 O | X |
---+---+---
   |   |

Pick an open slot:
5
That's not an open slot.

 O |   | X
---+---+---
 O | X |
---+---+---
   |   |

Pick an open slot:
7

 O |   | X
---+---+---
 O | X |
---+---+---
 X |   |

You win!
Press Enter to continue...
```