Select a group

In Moodle, please register to the group with the label that is written on your card. Make sure that you and your partner are in the same group.

The following exercises are recommended to be solved in the order presented. If you don’t reach the end, don’t worry – quality is more important than speed.

1. Nested loops

Let \( m \) be a list of lists, where the inner lists may have different lengths. For example,

\[
m = [[4, 3], [8, 2, 5, 1], [2]]
\]

The following program should print all elements of the list \( m \). Discuss with your partner and fill the blanks.

```python
for ___ in _________________:
    for ___ in _________________:
        print(__________)
```

Run this program in Thonny debugger. Make sure that your partner understands exactly the logic of this program and how it is executed. Ask questions, if you need clarification on something.

Let \( f(x) \) be some function that takes an integer \( x \) as its parameter and returns an integer. Let \( m \) be a list of lists. Assume that both \( f(x) \) and \( m \) are defined in the program. Write a piece of Python code that finds the element in list \( m \), for which the value of the function \( f(x) \) is maximal.

Make sure it is clear both to you and to your partner how this program works. Explain and ask questions, where needed.
2. Tic-tac-toe

Explain your solution of homework exercise 3 (number of wins in 4x4 tic-tac-toe game) to your partner until he/she fully understands it. Ask questions about your partner’s solution until you fully understand his/her solution.

Write, in your own words, what is the main idea of the algorithms used in both your programs.

Program 1:

Program 2:

Discuss with your partner and write below: what would be the best algorithm to solve this problem?

Now create a new program and implement a function `has_won`, which takes two parameters: a matrix that represents 4×4 tic-tac-toe game field and a symbol that is either 'X' or 'O'. The function returns the value `True`, if the given symbol has won the game, and `False` if the symbol has not won the game. Winning means having 3 consecutive symbols horizontally, vertically or diagonally.

For example, if

```python
matrix = [["O","","X","O"],
          ["O","X","","X"],
          ["X","","0",""],
          ["O","X","X","O"]]
```

then we get:

```python
>>> has_won(matrix, 'X')
True
>>> has_won(matrix, 'O')
False
```

*Hint*: use your homework program to check if the number of wins is greater than 0.

Next write a function `find_win` which takes two parameters: a matrix and a symbol. If the given symbol can win the game in one move, then the function prints out the game field after the winning move. If winning in one move is not possible, then the function prints out the message stating it.

```python
>>> find_win([["O","","X","O"],
              ["O","","","X"],
              ["X","","",""],
              ["0","X","X","O"]], 'X')
```
Player X can win:
O   X 0
0 X   X
X
0 X X 0

>>> find_win([['O',' ','X','O'],
            ['O',' ',' ','X'],
            ['X',' ',' ',' '],
            ['O','X','X','O']], 'O')

Player 0 cannot win.

Hint: look through all elements in the matrix using double loop. If the position is empty, make a copy of the matrix, add the symbol to this position and use the function has_won to check whether the symbol has won.

3. Line

There are \(n\) people standing behind each other in a line. Those persons who are taller than persons immediately before them, can also see other persons towards the beginning of the line, up to the person who is even taller. Write a program that reads the heights of persons standing in the line and finds the person who can see the farthest in the line. (In the following example all people look to the left.)

Enter heights: 195 167 165 190 172 182 187 189 168 174
Person 8 with height 189 can see the farthest.

4. Busses

A text file contains bus timetable between two cities. Each row contains departure time in the format hh:mm, arrival time in the same format and ticket price in euros; elements are separated by spaces. To travel from the first city to the second, it is obviously not useful to choose a bus that departs earlier, arrives later and costs more than some other bus. Write a program that prompts for a filename, reads the timetable and outputs all busses, in the order of departure times, that are useful for travelling from the first city to the second.

For example, if the content of timetable file is

<table>
<thead>
<tr>
<th>Time</th>
<th>Arrival</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00</td>
<td>11:30</td>
<td>5</td>
</tr>
<tr>
<td>10:00</td>
<td>13:00</td>
<td>6</td>
</tr>
<tr>
<td>09:15</td>
<td>12:15</td>
<td>7</td>
</tr>
<tr>
<td>09:30</td>
<td>12:00</td>
<td>6</td>
</tr>
<tr>
<td>10:15</td>
<td>12:45</td>
<td>5</td>
</tr>
</tbody>
</table>

then the program should output the following:

Enter filename: timetable.txt
Following busses are useful for travelling from the first city to the second:
09:00 11:30 5
09:30 12:00 6
10:15 12:45 5
5. Ant

In a square matrix, each element can be either 0 or 1. An ant is placed on a random cell of the matrix, initially facing north. The ant begins to move on the matrix according to the following rules. If the cell in which the ant currently is, contains 0, then the ant changes it to 1, turns left and moves one step forward to the neighbouring cell. If the cell contains 1, then the ant changes it to 0, turns right and again moves one step forward. If the ant steps over the border, it falls off the matrix.

At the beginning all elements are 0. Write a program that asks for the dimension of the matrix and, by repeatedly simulating this process, finds the average number of steps that the ant can do before it falls off the matrix. Also find the average percent of ones that are left in the matrix at the end.

Enter dimension of the matrix: 5
Average number of steps is [...]
Average percent of ones at the end is [...]

What do you think: how fast does the average number of steps increase, as the dimension of the matrix increases? Does the average percent of ones change, as the dimension increases, or stays approximately the same?