Week 4 worksheet

Name:

1. How do you prepare for practice sessions?

First, please answer to a small survey.

1. Each week’s material contains videos by Charles Severance. How exhaustively did you study the videos for this week? Please mark the cell that applies.

<table>
<thead>
<tr>
<th>I studied the videos thoroughly before continuing with homework problems</th>
<th>Quickly skimmed through the videos and reviewed them when necessary to solve the problems.</th>
<th>Quickly skimmed through the videos but didn’t rely on them very much while doing the homework.</th>
<th>Didn’t watch the videos at all.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Generally, how much have you watched lecture videos up to now?

3. How useful do you find these videos?

................. 1 - 5 (1 not at all useful, 5 very useful)

4. What other study materials have you used?

2. Bakery cash register

Compare your solution to the homework exercise 1 (home1.py) with the sample solution available in Moodle. Take either your solution or the sample solution as a basis.

The demand for cakes is lower on weekdays (Monday till Friday) but higher on weekends. So the bakery decides to decrease all prices of cakes on weekdays by 20% and to increase prices on weekends by 30%. Modify the definition of the function cake_price so that if the program is run on weekdays, it returns the price that is 20% lower than usual, and if it is run on weekends, then it returns 30% higher price.

Hint: see the datetime module (https://docs.python.org/3/library/datetime.html).

```python
>>> from datetime import date
>>> date.today().weekday()
6
```
3. Einstein's Theory of Special Relativity

Your homework program 2 (home2.py) contains the function \texttt{einsum}, which computes the sum of two speeds by Einstein's formula. Write the expression that uses the function \texttt{einsum} to compute the sum of the speeds 125000 and 275000:

Write the expression that uses the function \texttt{einsum} to compute the sum of two speeds that are stored in the variables \texttt{fastspeed} and \texttt{slowspeed}:

Write the expression that uses the function \texttt{einsum} to compute the sum of three speeds \(v_1\), \(v_2\), and \(v_3\):

Now write a program that contains the following functions:

- \texttt{add(number1, number2)}
  - Computes and returns the sum of two numbers.
- \texttt{multiply(number1, number2)}
  - Computes and returns the product of two numbers.
- \texttt{sumspeeds(speed1, speed2)}
  - Computes and returns the sum of two speeds using Einstein's formula but performing all additions and multiplications there by calling the functions \texttt{add} and \texttt{multiply}.

Write here the expression you used to compute the sum of speeds in your implementation of the function \texttt{sumspeeds}:

4. Budget

The Martin family is organising a party. They have invited a large number of people. Some guests have answered that they will come, others haven't yet but the Martins already need to estimate the expenses. The party budget consists of expenses on food (10 euros per person) and room rent (a fixed price of 55 euros regardless of the number of people).

First, create a function called \texttt{budget}. It has one parameter, the number of people, and calculates the budget size for given number people.

```python
>>> budget(8)
135
```
Next, write a program that:
1. prompts the user for the number of guests;
2. prompts the user for the number of guests who have answered;
3. calculates and outputs the maximum amount of the expenses if everyone attends (call the function);
4. calculates and outputs the minimum amount of the expenses if only those who have answered, show up (call the function).

How many people are invited? 26
How many have answered? 15
Maximum budget size = 315 EUR
Minimum budget size = 205 EUR

5. Pykkar

Pykkar is a virtual robot who operates in its virtual world. It can move, color squares, carry things, and has sensors to detect different objects. Pykkar is implemented in the module pykkar (this is not Python’s core module, so you probably need to install it: in Thonny select Tools > Manage packages...).

```python
from pykkar import *
create_world(""

########
#  >   #
#      #
#      #
#      #
#      #
########
"")
step()
step()
step()
right()
```

Pykkar understands the following commands:
- `step()` - Make a step forward.
- `right()` - Turn right 90°.
- `take()` - Pick up the cone that is on the next square. Pykkar can transport one cone at a time.
- `put()` - Put the cone on the next square. There can be up to 9 cones on a square.
- `push()` - Push the box or the cone that is on the next square.
- `paint()` - Paint the current square dark.
- `is_wall()` - Tell whether there is a wall in front (returns True or False).
- `is_cone()` - Tell whether there is a cone in front.
- `is_box()` - Tell whether there is a box in front.
- `is_painted()` - Tell whether the current square is painted.
- `get_direction()` - Tell the current direction (returns 'N', 'E', 'S', or 'W').
Pykkar’s world is created by the function `create_world`, which takes a multi-line string as a parameter. The string represents the map of the world, each symbol denotes one square. Types of square are:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(space)</td>
<td>Light floor tile</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>Dark floor tile</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Wall</td>
<td></td>
</tr>
<tr>
<td>^ &gt; v &lt;</td>
<td>Pykker on light tile, arrow indicates direction</td>
<td></td>
</tr>
<tr>
<td>N E S W</td>
<td>Pykker on dark tile, letter indicates direction</td>
<td></td>
</tr>
<tr>
<td>1 2 ... 9</td>
<td>Number of cones on light tile</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>One cone on dark tile</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>Box on light tile</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Box on dark tile</td>
<td></td>
</tr>
</tbody>
</table>

Exercises

a. Create a function that makes Pykkar turn left and a function to turn around.

b. Create a function that sets Pykkar’s view direction to north, regardless of its initial direction.

c. Create a function that makes Pykkar walk 5 steps but if it encounters the wall before completing, it stops without error.

d. Let the height and width of free area inside Pykkar’s world be \( m \) squares (3 \( \leq m \leq 6 \)) and \( n \) squares (2 \( \leq n \leq 6 \)), respectively. At the northeast corner there are \( n \) cones. The cones need to be transported to the south edge, placing one cone on each square there. Write a program that does this.

In the beginning:

```
#######
# >  5#
#    #
#    #
#    #
#    #
#######
```

In the end:

```
#######
# >    
#    #
#    #
#    #
#11111#
#######
```

At the end of session

Please submit your programs in Moodle.