The one with the popup market

Due 30 Sep by 23:59 **Points** 5 **Submitting** an external tool
Available until 30 Sep at 23:59

Successfully solving this problem will award the student **5 point** towards Examination 1

Requirements

- The script must pass all the automatic tests
- The script must pass a manual inspection
- **The student must write the script on his or her own**, any suspicion about cheating, cooperation, and/or plagiarism will be reported to the Discipline Board (see the **Study Guide** ↓ **(https://hkr.instructure.com/courses/5190/files/1027789/download?download_frd=1)** for more information). The students script will be checked both manually and using anti-plagiarism software

Instructions

1. Analyse the problem below using **Polya's Problem solving Technique** **(https://math.berkeley.edu/~gmelvin/polya.pdf)**
2. Create a script named **Lab1_5.py** that solves the problem according to the given instructions
3. Pay special attention when formatting the output as it must match the specified format **exactly** (character by character) to pass the tests
4. Upload the script using the link below

Task description

Last summer a relative of yours decided to start selling home grown vegetables at the local popup markets. In the beginning it was just for fun but it quickly became competitive and your relative has now asked you to help out by creating a script that use sales data to visually show what days are the most profitable. Your relative is very pedantic and want you to create the script according to the following instructions, special care must be taken to **exactly match the output** to the examples given.

When starting the script, the user shall be welcomed by a main menu that looks like this:

```
*----------------------------------------*
|               Main menu                |
*----------------------------------------*
| Option |           Description         |
*----------------------------------------*
|   1    | Register Purchase             |
|   2    | Register Repurchase           |
|   3    | Earnings (Amount per day)     |
|   4    | Earnings (Percentage per day) |
|   E    | Exit                          |
*----------------------------------------*
```

The different options will be described in more details below but first let us cover some general functionality of the script.

- All the sales data shall be entered manually as either a *Purchase* or a *Repurchase*, and the information displayed by the *Earnings* options shall update accordingly after each *Purchase*/*Repurchase*.
- After each action in the script (selecting an option in the menu) has completed, the user shall be returned to the main menu and be allowed to make another choice.
- The script shall give the following error message in case the user enters an invalid choice in any menu:

```
Error: Not a valid option!
```

- After an error message has been displayed (anywhere in the script), the user shall be sent back to the main menu.
- Even though the script has a strong focus on earnings, the script shall also keep track of if the user is making a profit or loss each day (see more details below).

Now, let us look at what each of the options in the main menu does in more detail.

Register Purchase

Selecting this option shall result in the following sub-menu being displayed.

```
*----------------------*
|       Purchase       |
*----------------------*
| Option | Description |
*----------------------*
|   1    |  Monday     |
|   2    |  Tuesday    |
|   3    |  Wednesday  |
|   4    |  Thursday   |
|   5    |  Friday     |
|   B    |  Go back    |
*----------------------*
```

In this menu the user selects for what day to register a purchase, and then the user shall be asked what amount to register. Note that the amount shall be added to any previous amount already registered for the same day. The user also has the option to go back to the main menu using the *Go back* option. Selecting an invalid option shall result in the error message seen earlier being displayed and the user being sent back to the main menu.

Register Repurchase

This option is very similar to the *Register Purchase* option in that the user shall select a day where to register the repurchase from a menu looking like this.

```
*----------------------*
|      Repurchase      |
*----------------------*
| Option | Description |
*----------------------*
|   1    |  Monday     |
|   2    |  Tuesday    |
|   3    |  Wednesday  |
|   4    |  Thursday   |
|   5    |  Friday     |
|   B    |  Go back    |
*----------------------*
```

After having selected the day the used shall enter the amount of the repurchase and deduct that amount from any previous amount registered to the selected day. Just as for the *Register Purchase* option the user shall be allowed to go back to the main menu without registering a repurchase, and selecting an invalid option shall result in the error message listed earlier being displayed and the user being sent back to the main menu.

Earnings (Amount per day)

If the user picks this option a bar chart of the total sales per day shall be displayed together with the actual numbers. The output shall follow the following format:

```
1000 |
     |
     |
     |
     |              *
 500 |  *           *   *
     |  *           *   *
     |  *   *       *   *
     |  *   *       *   *
     |  *   *       *   *
      --------------------
       Mo  Tu  We  Th  Fr

Monday    :        445kr
Tuesday   :        300kr
```

```
Wednesday :       -200kr (LOSS)
Thursday  :        567kr
Friday    :        465kr
```

There are a few things to note about the output that might not be obvious at first glance.

- The scale on the left-hand side shall be automatically adjusted to fit the registered data. The max value starts at 10 and is multiplied by 10 until the registered value for the day with the highest sale is less than or equal to the max value (max value starts at 10, then increases to 100, 1000, 10000…).
- Only the max value together with half the max value shall be displayed on the scale (note the alignment).
- The values covered by each star in the chart is relative to the calculated max value. If the calculated max value is 100kr, then each star would represent sales for an individual day according to the following (the star shall be drawn if the sales is higher than, or falls is the specified range):
  91 – 100kr
  81 – 90kr
  71 – 80kr
  61 – 70kr
  51 – 60kr
  41 – 50kr
  31 – 40kr
  21 – 30kr
  11 – 20kr
    1 – 10kr
- If the total sales for a particular day results in a negative amount, it shall be rounded up to zero when drawing the bar chart but the actual amount plus the tag (LOSS) shall be displayed when printing the actual amounts below the chart.

Earnings (Percent per day)

The output generated when the user selects this option is similar to the output generated by the *Earnings (Amount per day)* option, but it shows how many percent of the sales for the full week was done each day. The output shall be on the following format, showing both a bar chart and the actual percentages (with one decimal).

```
100% |
     |
     |
     |
     |
 50% |
     |              *
     |  *           *   *
     |  *   *       *   *
     |  *   *       *   *
      --------------------
       Mo  Tu  We  Th  Fr

Monday    :       25.0%
```

```
Tuesday   :        16.9%
Wednesday :         0.0%
Thursday  :        31.9%
Friday    :        26.2%
```

Take the following into account when creating the output.

- If the total sales for a particular day resulted in a negative amount, it shall be rounded up to zero when calculating the percentage for that day.
- The stars represent the percentage interval for each day according to the following (the star shall be drawn if the calculated percentage is higher than, or falls in the specified range):
  >90 – 100%
  >80 – 90%
  >70 – 80%
  >60 – 70%
  >50 – 60%
  >40 – 50%
  >30 – 40%
  >20 – 30%
  >10 – 20%
  > 0 – 10%

Exit
As the name hints at, selecting this option shall result in the script terminating.

Comments and additional resources

- Students are highly recommended to first solve all the tasks from the corresponding lectures and exercises (including the extra tasks) before attempting to solve this problem. Having a solid plan before starting to code will also help
- The student can upload as many times as he or she likes to before the deadline
- Calculate the percentages first when needed and use string formatting to round to one decimal when printing (for example using .1f)
- This **video** **(https://youtu.be/Xc_YX8m76L0)** recorded while running the finished script showcases how the script shall look and work when finished

This tool needs to be loaded in a new browser window

Load The one with the popup market in a new window