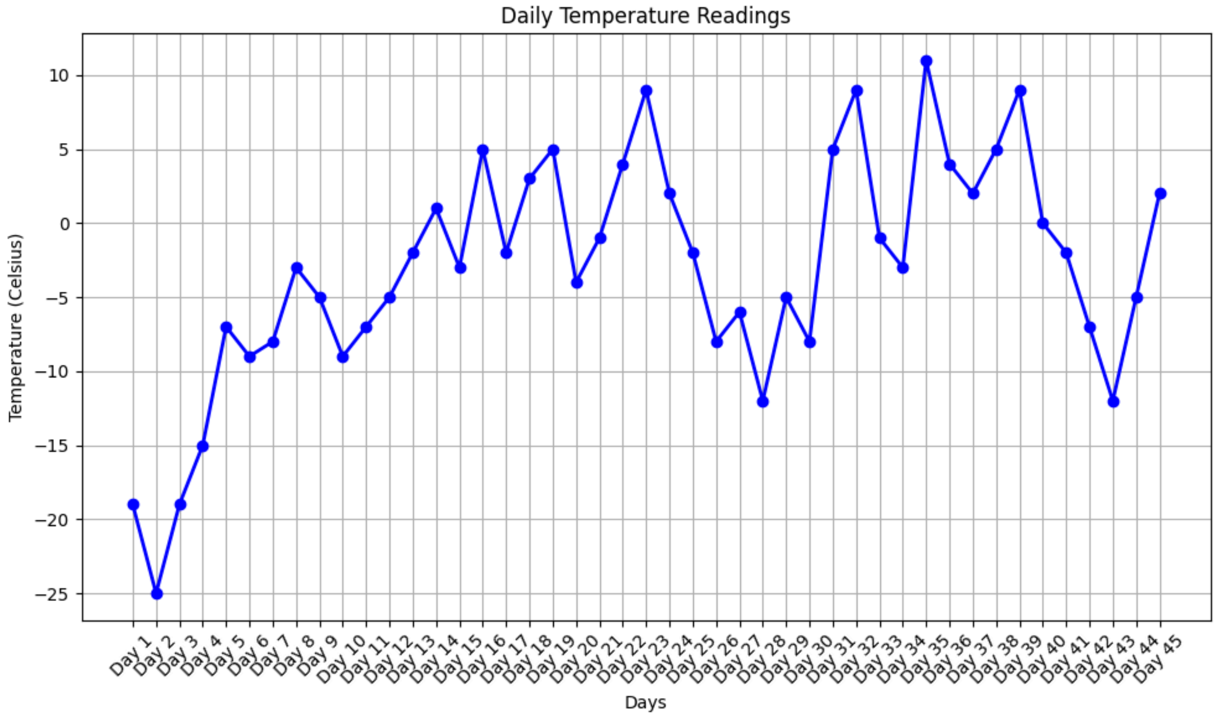
CENG 218 Spring 2024 Homework 1
Due date: 22nd of April 2024, 16:30

Note: Submit photos or scans of your solutions via MS-Teams, as well as bring hard copies to course assistants. Homeworks submitted after the due date will not be evaluated. Write all your explanations and comments in English! Text in Turkish will not be evaluated. Submit a single zip file that consists of a single pdf and a python code file. The file should be named as CENG218_HW1_<student_number>.zip (Omit the angle brackets). Cheating or collaborative work will not be tolerated and will be considered as zero grade.

Q1 (55 points)

Suppose that a meteorologist carries out a research about the climate change. She analyzed daily temperature data for a specific location over a year. An example of temperature reading data for forty days is shown in the image below. The goal is to identify the contiguous days with the highest sum of temperature to understand the temperature trend during that period.



She effortlessly finds the total temperature for a given period of time using an algorithm shown below.

```
TEMP(days, temperatures, start, end)
    start_index = start - days[1] + 1
    end_index = end – start + start_index
    temp = 0
    for d = start_index to end_index
        temp = temp + temperatures[d]
    return temp
```

Where *days* and *temperature* are the integer arrays; in addition *start*, *end* are the integers.
TEMP([14, 15, 16, 17, 18, 19, 20, 21, 22, 23], [-3, 5, -2, 3, 5, -4, -1, 4, 9, 2], 16, 21) returns 5.

The algorithm calculates the total temperature as (-2) + 3 + 5 + (-4) + (-1) + 4 = 5 between the 16th and 21st day including the end date. The complexity of this algorithm is $\Theta(d)$ where d is the number of days in a certain range. She tries to implement another algorithm that identifies the contiguous days with the highest sum of temperature as follows.

```
TARGET_TEMP(days, temperatures)
        ...
        ...
        ...
        return target_start, target_end
```

Example runs:
TARGET_TEMP(([22, 23, 24, 25, 26], [9, 2, 3, 11, 4]***)*** **returns** 22, 26.
TARGET_TEMP(([34, 35, 36, 37, 38], [-1, -3, 7, 4, 2]***)*** **returns** 36, 38.
TARGET_TEMP(([1, 2, 3, 4, 5], [5, -2, 3, 8, -4]***)*** **returns** 1, 4.

Note: Two integer arrays *days*, *temperatures* must be in the same length.

a. (20 pts) Design an algorithm to solve this problem and write its pseudocode as ***TARGET_TEMP(days, temperatures)***.
b. (20 pts) Analyze and define the asymptotically tight complexity by explaining the reason.
c. (15 pts) Implement the designed algorithm in the python (It will be named as *temperature_analysis.py*).

Q2 (45 points)
An algorithm running on an array determines 8 subarrays each of which is of length $\lfloor n / 2 \rfloor$ where n is the number of elements in the original array. The algorithm needs $\Theta(n)$ time in order to determine these subarrays. Then it calls itself for each of these subarrays.

a. (10 pts) Write the recurrence relation.
b. (15 pts) Use a recursion tree to determine asymptotic bound for the recurrence in (a).
c. (20 pts) Use substitution method to verify your answer in (b) as an asymptotic bound estimation. Do not forget to evaluate the base case for the solution.