

Bora Sozer

I confirm that this assignment is my own work.

Where I have referred to academic sources, I have provided in-text citations and included the sources in the final reference list.

Programming assignment

Part 1

Contents

[Weather Tracking App 2](#_Toc159843270)

[Section 1 :Algorithm 2](#_Toc159843271)

[Section 2: Technical Overview 3](#_Toc159843272)

[Section 3: Python Code with comments 4](#_Toc159843273)

[Section 4: Testing 6](#_Toc159843274)

[Section 5: Evaluation and Summary 8](#_Toc159843275)

[Section 6: References 9](#_Toc159843276)

[Event Booking App 9](#_Toc159843277)

[Section 1: Algorithm 9](#_Toc159843278)

[Section 2: Technical Overview 9](#_Toc159843279)

[Section 3: Python Code with comments 9](#_Toc159843280)

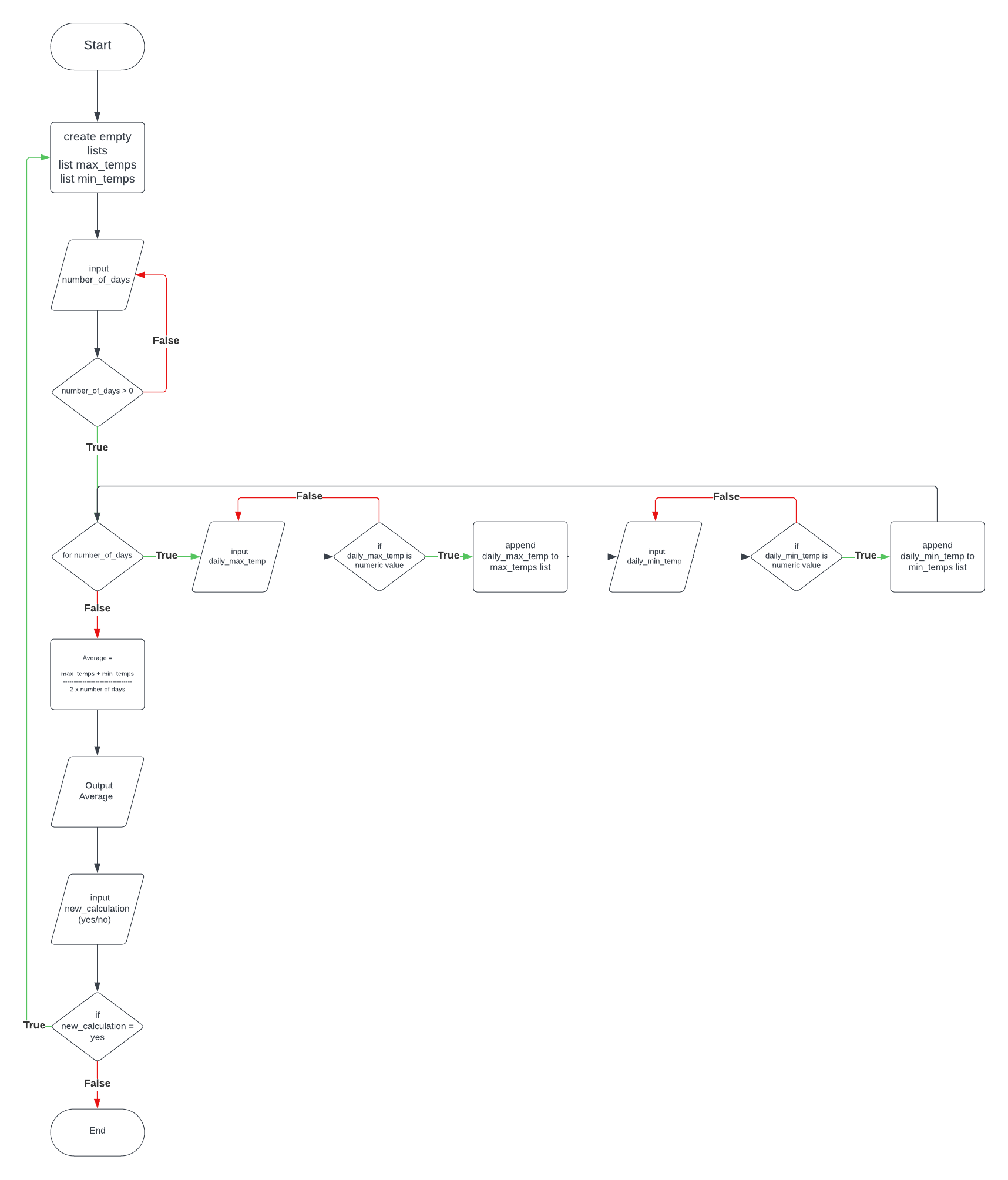
[Section 4: Testing 9](#_Toc159843281)

[Section 5: Evaluation and Summary 9](#_Toc159843282)

[Section 6: References 9](#_Toc159843283)

# **C:\Users\hakan\Desktop\cold-and-hot-icon-temperature-illustration-sign-thermometer-symbol-heat-logo-vector.jpg**Weather Tracking App

## Section 1 :Algorithm



## Section 2: Technical Overview

|  |  |  |
| --- | --- | --- |
| **Variables** | **Data structures** | **Functions** |
| days :  number of days to be calculated | max\_temps :  stores all daily max\_temp values | There is no defined function.(def()) |
| max\_temp :  daily maximum temperature value | min\_temps :  stores all daily min\_temp values | If-else :  Used if-else function to check whether the number of days selected is greater than 0 and in order to be able to make comments according to the calculated average |
| min\_temp :  daily minimum temperature value |  | While :  Used while function to get input again after errors, to repeat the code when the user chooses to perform a new tracking after the tracking |
| average\_max :  average of maximum temperature values (max\_temps / days) |  | For :  Used for function to receive input for the selected number of days |
| average\_min :  average of minimum temperature values (min\_temps / days) |  | Used list.append(value) function to add value to the list.  max\_temps.append(max\_temp)  low\_temps.append(low\_temp) |
| average\_overall :  average of maximum and minimum temperature values ((average\_max + average\_min) / 2) |  |  |

## 

## Section 3: Python Code with comments

while True:

#I stored the values in two lists because objective says "the program should calculate the average for both the highs and the lows over the set period"

max\_temps = []

min\_temps = []

#Learn number of days from user

while True:

#I used the try-except function to prevent a programme error when a value other than a number is entered.

try:

days = int(input("\nNumber of days:"))

#I wanted the number to be greater than 0 because we need at least 1 day to calculate it

if days <= 0:

print("Please enter a valid number of days (greater than 0).")

else:

break

except ValueError:

print("Please enter a valid number.")

print("")

#Input and store temperatures from user

for i in range(days):

while True:

#Showed user-friendly error messages and allow re-entry if inputs are invalid

try:

#I used a float function in case the temperatures might be fractional

max\_temp = float(input(f"{i+1}.day maximum temperature:"))

#I have limited the entry by taking the lowest and highest air temperature measured in the world

#because I wanted to make an program that could be used anywhere in the world

if -90 < max\_temp < 60:

max\_temps.append(max\_temp)

break

else:

print("Invalid temperature value. Please enter a temperature between -90 and 60 degrees Celsius.")

continue

except ValueError:

print("Please enter a valid value.")

while True:

try:

min\_temp = float(input(f"{i+1}.day minimum temperature:"))

if -90 < min\_temp < 60:

min\_temps.append(min\_temp)

break

else:

print("Invalid temperature value. Please enter a temperature between -90 and 60 degrees Celsius.")

continue

except ValueError:

print("Please enter a valid value.")

#Calculate and print average

average\_max = sum(max\_temps) / days

average\_min = sum(min\_temps) / days

average\_overall = (average\_max + average\_min) / 2

print(f"\nAverage max temperature:{average\_max:.1f}")

print(f"Average overall temperature:{average\_overall:.1f}")

print(f"Average min temperature:{average\_min:.1f}\n")

#Comment based on the average temperature.

#I tried to interpret the average both for myself and for the UK.

if average\_overall > 35:

print("It's extremely hot on average.")

elif average\_overall > 30:

print("It's quite hot on average.")

elif average\_overall > 25:

print("It's warm on average.")

elif average\_overall > 20:

print("It's moderately warm on average.")

elif average\_overall > 15:

print("It's mild on average.")

elif average\_overall > 10:

print("It's cool on average.")

elif average\_overall > 5:

print("It's cold on average.")

elif average\_overall > 0:

print("It's very cold on average.")

elif average\_overall > -5:

print("It's extremely cold on average.")

elif average\_overall > -10:

print("It's dangerously cold on average.")

else:

print("It's deadly cold on average.")

#Learn from the user whether to continue the calculation or exit the app

new\_calculation = input("\n1-New calculation\n2-Quit\nType number:")

while not new\_calculation in ["1","2"]:

new\_calculation = input("\nInvalid choice. Write 1 or 2:")

if new\_calculation == "2":

exit()

## Section 4: Testing

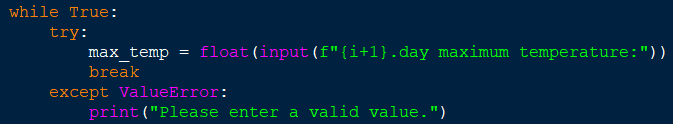
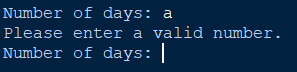
**Testing for development:**

Problem 1:

In order to collect the values entered as temperature, it must be converted to integer. Python int() or float()function was giving an error when an input other than a number was made.(Figure 2)

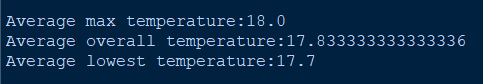
C:\Users\hakan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_8.png(Figure 2)

I solved this problem as Figure 2-3 with the try-except function that I searched and found from “pythonbasics.org/try-except/”.

 **Figure 2.** **Figure 3.**

Problem 2:

Since I also share the average result with the user, if the result was long fractional, the output was not user-friendly and unnecessary.

****

I solved this by displaying a digit after the comma using “.1f “

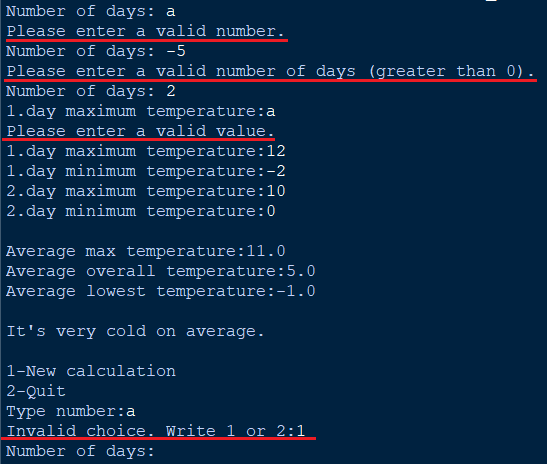
“.1f “is a format specifier that formats floating-point numbers with one decimal place. This means it displays only one decimal place after the decimal point.



C:\Users\hakan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_11.png

**Testing for evaluation:**

Program was tested for every possibilities. I have tried with different inputs to make the programme give an error, but thanks to the functions I use, the programme can solve the problem without giving an error.



|  |  |  |
| --- | --- | --- |
| Purpose of test | How the test was carried out | Outcome from that test |
| Validate user input for number of days  Ensure the program does not accept negative or non-numeric values | Negative number and non-numeric values tried as input from user | Program warned the user that they had typed invalid and asked them to type again |
| Validate user input for temperature values  Ensuring that only numeric input is accepted and that temperatures are within reasonable ranges | Non-numeric values and unreasonable temperature input tried | Program warned the user that they had typed invalid or unreasonable and asked them to type again |

## Section 5: Evaluation and Summary

Programme fulfils all the desired conditions and successfully fulfils the objective.

It receives input data from the user in a simple way. It validate user inputs, ensuring they are numeric values. If not it output user-friendly warning message and wants re-input.

Stores all values and calculates the overall average temperature. Makes comments to the user according to the overall average temperature.

Outputs the average of the daily maximum, minimum, overall degrees. If there are too many fractions, it rounds and leaves a single fraction.

The programme is in such a way that everyone can understand and use it. Input and output parts are user friendly.

All vulnerabilities that would cause the programme to crash have been closed. Also checked in tests and passed successfully.

If the programme is developed in the future, it can make user-based comments at an advanced level. For example, when the application is opened, a question can be asked to measure the user's temperature perception and comments can be made accordingly. It checks that the number of days is greater than 0 and temperature is within reasonable values

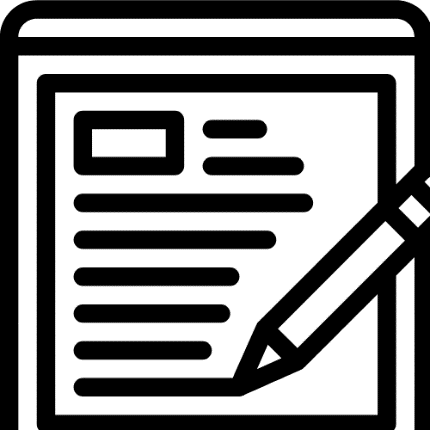
## Section 6: References

LucidChart for Weather Tracking App: https://lucid.app/lucidchart/96a4a116-01ea-475d-b4dd-1b95abbce24a/edit?invitationId=inv\_d328f64b-71a6-4913-84bc-ffb52b48d285

Try-except error solving <https://pythonbasics.org/try-except/>

<https://www.w3schools.com/python/python_datetime.asp>

<https://docs.python.org/3/library/datetime.html>

****

# Event Booking App

## Section 1: Algorithm

## Section 2: Technical Overview

## Section 3: Python Code with comments

## Section 4: Testing

## Section 5: Evaluation and Summary

## Section 6: References