The 1D array `cityName[]` contains the name of cities in a country. The 2D array `cityTemperature[]` includes the temperatures for each city, recorded every day at 12 pm for a single Month as type `REAL`.

The position of each city's data in the two arrays is the same; for example, the city in position 5 in `cityName[]` and `cityTemperature[]` are the same. The index Day denotes the day of the month in which the temperature is recorded.

The variable `numOfDays` contains the total number of days in the month, and the variable `numOfCities` contains the total number of cities in the array. The variables and the arrays have already been defined.

Write a program that completes the following tasks:

- Calculates a city's average temperature over the month, rounded to the nearest whole number.
- Outputs the following for each city
    - Name of city
    - The average temperature over the month
    - Day on which the temperature is highest over the month.
- Creates a file called `highest.txt` and writes to it the name of city with the highest average temperature and the day that it happened.
- All outputs must be formatted appropriately.

You must use pseudocode or program code and add comments to explain how your code works.

You do not need to initialise the data in the array or define the variables set in the question.

Python Codes

```
# Assuming cityName, cityTemperature, numOfDays, and numOfCities are predefined

# Initialize variable for hottest average temperature to 0
HottestAve = 0
HottestCity = ""

# Iterate over the number of cities
for row in range(numOfCities):  # Adjusting for Python's zero-based indexing
    HottestTemp = 0
    HottestDay = 0
    TotalTemp = 0

    # Iterate over the number of days
    for col in range(numOfDays):  # Adjusting for Python's zero-based indexing
        TotalTemp += cityTemperature[row][col]
        if cityTemperature[row][col] > HottestTemp:
            HottestTemp = cityTemperature[row][col]
            HottestDay = col + 1  # Adjust for human-readable day indexing

    # Display the city name
    print("City name:", cityName[row])

    # Display the average temperature rounded to 0 decimal places
    average_temp = round(TotalTemp / numOfDays)
    print("Average temperature:", average_temp)

    # Display the day with the highest temperature
    print("Day with the highest temperature:", HottestDay)

    # Check if the current average is the hottest average, update if it is
    if average_temp > HottestAve:
        HottestAve = average_temp
        HottestCity = cityName[row]

# Write the city with the hottest average temperature and the average to a file
with open("highest.txt", "w") as file:
    file.write(f"{HottestCity}\n{HottestAve}")
```

Pseudocode

```
// Initiate variable for hottest average temperature to 0
HottestAve ← 0

//Iterate the number of cities (row)
FOR row ← 1 TO numOfCities

   //Set total temperature, hottest temperature to 0 and hottest day to 0
   // reset at every loop
   HottestTemp ← 0
   HottestDay ← 0
   TotalTemp ← 0

   //Iterate number of days (column)
   FOR col ← 1 TO numOfDays
      TotalTemp ← TotalTemp + cityTemperature[row,col]
      // Check if the daily temperature is larger than the hottest temperature
      IF cityTemperature[row,col] > HottestTemp
         THEN
            // find city with the hottest temp
            HottestTemp ← cityTemperature[row,col]
            HottestDay ← col
      ENDIF
   NEXT col

      // Display the city name
      OUTPUT "City name:",cityName[row]

      // Display the average temperature rounded to 0 decimal places
      OUTPUT "Average temperature: ", ROUND(TotalTemp/numOfDays,0)

      //Display the hottest day for that city
      OUTPUT "Day with the highest temerature: ",HottestDay

   // Check if the average is the hottest average,
   IF ROUND(TotalTemp/numOfDays,0) > HighestAve
         // find city with the hottest average
         HottestAve ← ROUND(TotalTemp/numOfDays,0)
         HottestCity ← cityName[row]     , HighestAve ← ROUND(TotalTemp/numOfDays, 0)
      ENDIF
   NEXT row


   // write hottestCity and the average to file
   OPENFILE highest.txt FOR WRITE
   WRITEFILE highest.txt, HottestCity
   WRITEFILE highest.txt, HottestAve
   CLOSEFILE highest.txt
```

Marking Instructions in italics			
AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems			
0	**1-3**	**4-6**	**7-9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check the list of techniques needed.*	The range of programming techniques used is appropriate to the problem. *All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

In writing your Pseudocodes observe these rubrics with the equivalent number of points marks. Aim for the top marks.

Marking Instructions in italics			
AO3: Provide solutions to problems by: • **evaluating computer systems** • **making reasoned judgements** • **presenting conclusions**			
0	**1-2**	**3-4**	**5-6**
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented.
	Some identifier names used are appropriate. *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All of the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario.*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution attempts to meet most of the requirements. *Solution contains lines of code that perform most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*