Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Summer 2020 - 2021

Lab Guide #1 – Week 1 - 3

OBJECTIVE : General Review of 151 Subjects

Instructors : Okyay SAY
Assistants : Ruşen ASAN

Q1. Write a C program that will take a positive number as input, and output the digits of each given number in **reverse** order. Your program must check for validity of the given input and then it must give an error message if the number is not a positive number.

Example Run:

```
Enter a positive number: -3
Sorry! You didn't give a positive number!!!

Enter a positive number: 0
Sorry! You didn't give a positive number!!!

Enter a positive number: 260317
  7  1  3  0  6  2
```

Project Name: LG1_Q1
File Name: Q1.cpp

Q2. Write a C program that gets values of time in minutes as whole numbers until a sentinel value of -1 has been given; in order to calculate the given values' respondent days, hours and minutes.

Write the following function;

- **formatTime** that gets a duration in minutes as a parameter and returns this time duration in days, hours and minutes.

Example Run:

```
Enter a duration : 69753
Duration is 48 days, 10 hours and 33 minutes

Enter a duration : 20450
Duration is 14 days, 4 hours and 50 minutes

Enter a duration : 24
Duration is 0 days, 0 hours and 24 minutes

Enter a duration : 150
Duration is 0 days, 2 hours and 30 minutes

Enter a duration : -1
```

Project Name: LG1_Q2
File Name: Q2.cpp

Q3. The text file "**numbers.txt**" contains 100 integer numbers. You will generate the random number ***n*** between 10 and 100 and read ***n*** numbers from the file into a one-dimensional integer array with the maximum size 100. You will find and display the average of these numbers.

Write the following functions;

- **findAvg** that takes an integer array and the number of elements as input parameters, finds and returns the average of the numbers in the array.
- **display** that takes an integer array and the number of elements as input parameters, displays the content of the array on the screen.

HINT to generate a random number: Do not forget to include <stdlib.h> and <time.h> libraries.
srand(time(NULL)); `// Use current time as seed for random generator`
randomNum = rand(); `// Create same sequence of random numbers on every program run`

Example Run:

```
Generated number is 19

The array content is;
  54   28   36   33   57   47   77   81   31    8
   1   20   47   56   12    9   14    3   25

The average of these numbers are 33.63
```

numbers.txt

```
54 28 36 33 57 47 77 81 31 8 1 20 47 56 12 9 14 3 25 78
15 43 18 2 19 44 11 4 89 8 97 5 27 34 55 45 12 84 64 11
6 20 24 82 10 8 11 22 28 9 19 46 55 83 30 7 35 74 33 54
27 98 11 32 56 46 76 1 18 5 63 10 5 19 23 81 9 7 10 21 3
88 7 96 4 26 75 82 11 8 13 2 24 76 14 42 17 53 27 9 12
```

Project Name: LG1_Q3
File Name: Q3.cpp

Q4. Write a C program that specifying the winner of a voice contest according to the audience votes. There are **5** competitors and the audience gives a vote for a competitor using their competitor number. The votes are stored in the "**votes.txt**". The program counts the number of votes and displays a table that shows the frequency distribution of each competitor as shown in the example run below.

Write the following function;

- **findMaxFreq** that takes the frequency distribution array as parameter, finds and returns the index of the winner of the competition.

Example Run:

```
Competitor     Frequency
   1              5
   2              4
   3              8
   4              4
   5              2

The winner is 3. competitor...
```

votes.txt

```
3 1 4 5 3 2 3 1 3 4 2 3 1 3 2 3 4 1 3 5 4 2 1
```

Project Name: LG1_Q4
File Name: Q4.cpp

Q5. You have been given the records of about annual temperature averages of some provinces in Turkey, which had been acquired between the years 2016 and 2020 in the "**Q5.txt**" file. The first column of this text file has the codes of each of these provinces from which the data was gathered and the rest of its columns have the temperature averages for the respective years.

Write a C program that reads the file **"Q5.txt"**, stores these province codes and the corresponding temperature averages in parallel arrays, through which the following must be calculated:

- The highest average degree recorded in all years and in all provinces.
- The averages of every province's 5 year long records.

Write the following functions;

- **readFile** that takes a file pointer and the two arrays to be filled, in order to read the file content and store them into the necessary arrays of codes and averages.
- **findHighestDegree** that takes a two-dimensional array and pointers as references of some variables, then, it returns row and column subscript numbers of the highest integer value among the array elements.
- **findTotals** takes two arrays and calculates the totals for each row of the two-dimensional array's into the one-dimensional array in order.

Q5.txt

```
07 28 30 32 37 29
06 25 20 30 35 24
34 17 21 12 8 15
35 22 21 16 22 11
```

Province Codes

Years: 2016, 2017, 2018, 2019, 2020

output.txt

```
Average highest degree is: 37
Year: 2019
City: 07

City   Total
07     156
06     134
34     73
35     92
```

Project Name: LG1_Q5
File Name: Q5.cpp

Additional Questions

AQ1. Write a C program that takes several integers until a non-positive integer is entered and displays a hexagon using the function.

Write the following function;

- **drawHexagon** that takes an integer n as a parameter and displays a hexagon with n lines as in the example run.

Example Run:

```
Enter an integer (<=0 to stop): 5
    ***
   *****
  *******
 *********
***********
 *********
  *******
   *****
    ***

Enter an integer (<=0 to stop): 3
  ***
 *****
*******
 *****
  ***

Enter an integer (<=0 to stop): -1
```

Project Name: LG1_AQ1
File Name: AQ1.cpp

AQ2. Write a C program that reads IPINs and tennis scores of several tennis players from the file tennis.txt; finds and displays the average of each tournament and the average of each player using the functions below. See the example run.

Write the following functions;

- **readFromFile** that takes a file pointer, a one-dim array to keep the player IPINs and a two-dimensional array to keep the tournament scores as parameter. The function reads the player IPINs into the one-dim array and 4 game scores of several tennis players into the two-dim array from the specified file. The function also returns the number of players.
- **findPlayerAvg** that takes the two-dim scores array and the number of player as input parameters, finds the average of each player and stores the averages into a one-dim array.
- **findTournamentAvg** that takes the two-dim scores array and the number of player as input parameters, finds the average of each tournament and stores the averages into a one-dim array.
- **displayTournamentAvg** that takes the one-dim array which keeps the tournament averages as input parameter and displays the averages of all tournaments on the screen.

Example Run:

```
Player IPINs  Average
************  *******
 11           481.75
 22           500.00
 33           480.25
 44           434.00
 55           453.75
 66           461.50
 77           519.25
 88           525.00
 99           476.75
 12           485.00

Tour Number  Average
***********  *******
    1         477.1
    2         483.6
    3         505.6
    4         460.6
```

tennis.txt:

```
11 475 570 500 382
22 450 550 575 425
33 375 482 552 512
44 352 545 314 525
55 560 385 475 395
66 496 520 345 485
77 373 582 698 424
88 545 510 570 475
99 595 347 465 500
12 550 345 562 483
```

Project Name: LG1_AQ2
File Name: AQ2.cpp

Debugger Short Keys

Display documentation for the active window.	F1
Display a system menu for the application window.	ALT+SPACEBAR
Add and remove breakpoints on the current lines.	**F9**
Clear all breakpoints.	CTRL+SHIFT+F9
Disable breakpoint.	CTRL+F9
Display Breakpoints dialog box.	CTRL+B
Adds a watch on the currently selected word.	SHIFT+F9
End debugging session.	**SHIFT+F5**
Execute code one statement at a time, following execution into function calls (Step Into).	**F11**
Execute the next line of code but without following execution through any function calls (Step Over).	**F10**
Execute the remaining lines of a function in which the current execution point lies (Step Out).	SHIFT+F11
Restart a debugging session.	CTRL+SHIFT+F5
Resume execution of your code from the current statement to the selected statement (Run to Cursor).	CTRL+ F10
Run the application to the next break point.	**F5**
Set the next statement.	CTRL+SHIFT+ F10
Stop execution (Break).	CTRL+BREAK
Open watch window #x.	**CTRL+ALT+W, #{1, 2, 3}**
Open locals' window.	**CTRL+ALT+V, L**