Department of Information Systems and Technologies

CTIS151 – Introduction to Programming

SPRING 2023 - 2024

Lab Guide #18 – Week 14 - 2

OBJECTIVES : Two-dimensional Arrays as input/output function parameters

Instructors : Burcu LİMAN
Assistants : Engin Z. KIRAÇBEDEL, Sıla YAPICI

Q1. Write a C program that reads IDs and 4 parts of IELTS exam (Reading, Listening, Speaking and Writing) of several students from the file named **ielts.txt** to calculate the average of grades. If the average is greater than 6.5, the student will be eligible, not eligible, otherwise. Display the exam info shown in the example.

Write the following functions;

- **readFromFile:** takes the file pointer, one-dimensional **id** array and two-dimensional the **points** array as parameters, reads the id and grades for Reading, Listening, Speaking and Writing into the arrays. The function should return the actual number of students.
- **display:** takes the one-dimensional **id** array, two-dimensional the **points** array and the number of students as parameters, displays the student information with the average grade and the eligibility of the student (Y/N).

Project Name: LG18_1
File Name: Q1.cpp

Example Run:

```
ID      R       L       S       W       ELIGIBLE        OVERALL
-----------------------------------------------------------------
111     1.00    2.00    1.00    4.00     N              2.00
112     6.50    6.50    6.50    6.50     Y              6.50
5       6.50    6.00    5.00    8.00     N              6.38
2008    4.00    5.50    6.50    7.00     N              5.75
563     6.00    7.00    8.00    9.00     Y              7.50
```

ielts.txt

```
111     1       2       1       4
112     6.5     6.5     6.5     6.5
5       6.5     6       5       8
2008    4       5.5     6.5     7
563     6       7       8       9
```

Q2. Write a C program that reads IDs and game scores of several bowling teams from the file **"bowling.txt";** finds and displays the average of each game and the average of each team using the functions above. See the example run.

Write the following functions;

- **`readFromFile`:** takes a file pointer, a one-dim array to keep the team Ids and a two-dimensional array to keep the game scores as parameter. The function reads the team IDs into the one-dim array and 4 game scores of several bowling teams into the two-dim array from the specified file. The function also returns the number of teams.
- **`findTeamAvg`:** takes the two-dim scores array and the number of team as input parameters, finds the average of each team and stores the averages into a one-dim array.
- **`findGameAvg`** takes the two-dim scores array and the number of team as input parameters, finds the average of each game and stores the averages into a one-dim array.
- **`displayGameAvg`:** takes the one-dim array which keeps the game averages as input parameter and displays the averages of all games on the screen.

Project Name: LG18_Q2
File Name: Q2.cpp

Example Run:

```
Team Number  Average
***********  *******
 12          483.50
 24          436.25
 33          505.25
 45          470.00
 57          517.50
 68          449.00
 79          444.25
 89          500.00
 96          484.00
 98          455.50

Game Number  Average
***********  *******
1            475.7
2            482.1
3            496.0
4            444.3
```

bowling.txt

```
12 482 570 500 382
24 350 395 575 425
33 475 482 552 512
45 552 545 418 365
57 660 385 475 550
68 446 520 345 485
79 273 582 498 424
89 445 510 570 475
96 624 347 465 500
98 450 485 562 325
```

Q3. Write a C program that reads several characters from a text file named **"words.txt"** into a two-dim array. Get the number as a dimension and display the major and minor diagonal of the square matrix in the given file.

Write the following functions:

- **`readFromFile:`** gets the input file pointer and two-dim char array to find the original square dimensional.
- **`displayTwoDimension:`** gets a two-dim char array and its square size to display its contents.
- **`displayDimension:`** gets a two-dim char array and dimension as parameters, displays the characters on the major and minor diagonal of the square matrix.

Example Run:

```
bkltgdrfm
nurcaglar
kytlolsak
1o1tvcert
kasseport
xxxlxrxxx
dfiderfde
3n3s567lk
ekranabay

Enter the dimension:[1-9] 46
Enter the dimension:[1-9] 9
Dimension for 9, MAJOR elements of the file is: butterfly
Dimension for 9, MINOR elements of the file is: masceline

Enter the dimension:[1-9] 6
Dimension for 6, MAJOR elements of the file is: butter
Dimension for 6, MINOR elements of the file is: dal1ax

Enter the dimension:[1-9] 3
Dimension for 3, MAJOR elements of the file is: but
Dimension for 3, MINOR elements of the file is: luk

Enter the dimension:[1-9] -1
EXIT!
```

words.txt

```
bkltgdrfm
nurcaglar
kytlolsak
1o1tvcert
kasseport
xxxlxrxxx
dfiderfde
3n3s567lk
ekranabay
```

Project Name: LG18_Q3
File Name: Q3.cpp

ADDITIONAL QUESTION

You will provide a software for a company which reads the stock information that includes ***product id, price, number of product in the stock**,* from the **company.txt** file, and simulates a customer's shopping and prepares a receipt for the customer writing the details into the **shopping.txt** file.

Write a C program that display a menu on the screen until user selects the EXIT option. Then the program reads the stock information into a two dimensional array and according to the choice from the menu it displays List of Items in the stock or customer will buy a product. Also finally display the total payment on the console.

Write the following functions;

- **menu:** displays the menu, reads and returns the user's choice. Make a data validation for the choice.
- **listStockInfo:** takes the two-dimensional **company array** as input parameter and displays the content of the array.
- **searchProd:** takes the two-dimensional **company array** and the **product id** to be searched as input parameters, then searches the array and returns **the index of the product**. Otherwise returns -1.
- **buyProd:** takes the two-dimensional **company array** and **output file pointer** as input parameters and the **sum of the payment** as output parameter.

  The function will read the id and the quantity to be bought from the user by validating the values, calculates the payment for the product and write the **product id, price, quantity** and the **payment for the product** to the given file.
  The function also calculates and returns the **total payment**.

  DO NOT forget to validate and display warning messages for the product id and quantity. See the example run.

company.txt

```
111 88 200
122 120 32
123 166 5
202 30 11
356 390 6
117 260 2
288 80 100
242 45 1
130 166 39
333 275 12
345 490 4
120 299 407
```

shopping.txt

```
ID      PRICE  QUANTITY  PAYMENT
------  -----  --------  -------
   123    166         5      830
   130    166         2      332
   120    299         3      897

Total payment of the customer is: 2059 TL
```

Project Name: LG18_AQ
File Name: AQ.cpp

Example Run :

```
 MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:1

    ID Price Stock
------ ----- -----
   111    88   200
   122   120    32
   123   166     5
   202    30    11
   356   390     6
   117   260     2
   288    80   100
   242    45     1
   130   166    39
   333   275    12
   345   490     4
   120   299   407

MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:2

Enter the product id: 555

Wrong product number
Re-Enter the product id: 123
Enter the quantity: 6

There are 5 product in the stock
Re-Enter the quantity: 10

There are 5 product in the stock
Re-Enter the quantity: 5
```

```
MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:1
    ID Price Stock
------ ----- -----
   111    88   200
   122   120    32
   123   166     0
   202    30    11
   356   390     6
   117   260     2
   288    80   100
   242    45     1
   130   166    39
   333   275    12
   345   490     4
   120   299   407

MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:2

Enter the product id: 123
SORRY! The product is out of stock

MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:2

Enter the product id: 130
Enter the quantity: 2
```

```
MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:120
Enter choice:2

Enter the product id: 120
Enter the quantity: 3

MENU
1. List Stock Info
2. Buy a product
3. Exit
Enter choice:3
    ID Price Stock
------ ----- -----
   111    88   200
   122   120    32
   123   166     0
   202    30    11
   356   390     6
   117   260     2
   288    80   100
   242    45     1
   130   166    37
   333   275    12
   345   490     4
   120   299   404
```