Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms
Spring 2024 - 2025

Lab Guide #6 - Week 4 - 1

OBJECTIVE : Structure and dynamic memory allocation exercises

Instructor : Serpil TIN
Assistants : Berk ÖNDER & Hatice Zehra YILMAZ

The %s operator is one used for reading strings of characters in to character arrays using the scanf function.

Q1. A hotel needs software to keep the customers' information: **name, birth year, day, payment,** and **room info** including **room number and room type**. USE **dynamically** created NESTED structure array with ***n*** customers (for n generate a **random** value between **5** and **20**).

Write the following functions;

- **`readFromFile`** : reads the customers' information from the file "**customers.txt**" into a dynamically created structure array, calculates the payment based on the number of days and the room type (see the table below for the room prices), and stores the results in the payment field of each customer in the structure array.

Room Type	Price Per Night
S - Single	1000 TL
T - Twin	2500 TL
D - Double	3500 TL

- **`displayAll`** : displays the information of all customers as in the example run.
- **`calculateTotalEarnings`**: calculates the total earnings of the hotel and returns it.
- **`findCustomerMaxPayment`**: finds and returns the index of the customer paying the maximum price for the accommodation.

Write a C program that reads and displays the customers' information. The program will also do the following;

- Displays the total earnings of the hotel,
- Displays the customer information paying the price for the accommodation,

Please examine the given example run, and test your programs.

Project Name: LG6_Q1
File Name: Q1.cpp

Example Run#01:

```
9 Customers' info will be read:

Name           BYear   Day   Room Num   Type    Payment
-----------------------------------------------------
AlexSmith       1987     5    5008        S     5000.00
EmmaJohnson     1995     3    5112        T     7500.00
DanielWilliams  1972     2    5004        S     2000.00
OliviaJones     1988     1    5010        S     1000.00
AvaDavis        1999     4    5140        T    10000.00
LiamMiller      1983     2    5030        S     2000.00
SophiaWilson    1978     2    5205        D     7000.00
NoahMoore       1965     1    5035        S     1000.00
MiaTaylor       1990     1    5130        T     2500.00

Total Earnings of the hotel : 38000.00

The Customer having maximum payment for accommodation:
AvaDavis with the price 10000.00 TL
```

customers.txt

```
AlexSmith 1987 5 5008 S
EmmaJohnson 1995 3 5112 T
DanielWilliams 1972 2 5004 S
OliviaJones 1988 1 5010 S
AvaDavis 1999 4 5140 T
LiamMiller 1983 2 5030 S
SophiaWilson 1978 2 5205 D
NoahMoore 1965 1 5035 S
MiaTaylor 1990 1 5130 T
LucasAnderson 1986 4 5050 S
EthanBrown 1976 8 5210 D
IsabellaJackson 1971 3 5120 T
OliverWhite 1994 8 5025 S
HarperHarris 1977 4 5215 D
JackMartin 1989 2 5040 S
AmeliaLee 1975 1 5115 T
AidenNelson 1969 5 5045 S
GraceCooper 2002 2 5150 T
LoganWard 1960 3 5060 S
LilyFisher 1963 7 5145 T
```

Example Run#02:

```
15 Customers' info will be read:

Name            BYear   Day   Room Num   Type     Payment
-----------------------------------------------------------
AlexSmith        1987     5     5008         S     5000.00
EmmaJohnson      1995     3     5112         T     7500.00
DanielWilliams   1972     2     5004         S     2000.00
OliviaJones      1988     1     5010         S     1000.00
AvaDavis         1999     4     5140         T    10000.00
LiamMiller       1983     2     5030         S     2000.00
SophiaWilson     1978     2     5205         D     7000.00
NoahMoore        1965     1     5035         S     1000.00
MiaTaylor        1990     1     5130         T     2500.00
LucasAnderson    1986     4     5050         S     4000.00
EthanBrown       1976     8     5210         D    28000.00
IsabellaJackson  1971     3     5120         T     7500.00
OliverWhite      1994     8     5025         S     8000.00
HarperHarris     1977     4     5215         D    14000.00
JackMartin       1989     2     5040         S     2000.00

Total Earnings of the hotel : 101500.00

The Customer having maximum payment for accommodation:
EthanBrown with the price 28000.00 TL
```

GENERATION OF RANDOM NUMBERS:

1. Use stdlib.h (for srand function)
2. Use time.h (for time function).
3. srand(time(0)); for getting different number every time you run the program.
4. For getting a random number between 0 – 50: num = rand() % 51;

```
// to create a number between a range
//num = rand() % ((MAX+1)- MIN) + MIN
```

5. Apply debug process to check the random number.

Example program:

```
#include <stdio.h>
#include <stdlib.h> //for srand funtion
#include <time.h>  //for time function


int main(void)
{
    int num;
    /* we use srand function to be able to get a random number but we cannot use the srand function on its own, so we
    also use time function in it to give a start point to the srand function; because time is different every time you
    run the program, the random number will be different also */
    srand(time(NULL));


    /* because time returns a very big number it returns the millisecond value of the hour, so we want to get a random
    number between 0 and 99, we get the modulus 100 of the rand      function */
    num = rand() % 100;


    /* to create a number between a range*/
    //num = rand() % ((Max+1)-Min) + Min
    printf("The random number is: %d", num);


    return 0;
}
```

Example Run #1:

```
The random number is: 99
```

Example Run #2:

```
The random number is: 26
```

Q2. A fitness center needs a software program to monitor members' workout progress. Each member's information consists of **member ID, name,** and **workout details**, including exercise type ('**C**': Cardio, '**S**': Strength Training, '**Y**': Yoga), hours spent on that exercise in a week, and calories burned per hour.

The text file "**fitness.txt**" keeps the number of members on the first line. The following lines contain the member details.

Write a C program that reads the members' information from the text file into a **dynamically** created NESTED structure array, displays a menu, and does the necessary operations according to the user's choice. The program stops when the user wants to exit.

Write the following functions that;

- **`menu:`** displays the following menu, reads, validates, and returns the user's choice.

```
FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
```

- **`readFromFile:`** takes the file pointer, members structure array, and the number of members as parameters. The function reads the members' information from the file into the array.
- **`displayMemberInfo:`** takes the members array and the number of members as parameters and displays all members's information by calculating total calories burned.

$$Total\ Calories\ Burned = hoursSpent \times caloriesBurnedPerHour$$

- **`searchMember:`** takes the members array, the number of members, and the member ID to be searched in the array as parameters. Then, searches for the member in the array and, if the ID exists, returns the index of the member. Otherwise, it returns -1.
- **`mostActiveMem:`** takes the members array and the number of members as parameters, finds the most active member based on the workout hour, and returns his/her index.
- **`mostPopEx:`** takes the members array and the number of members as parameters, calculates the total hours spent on each exercise type, finds and displays the most popular exercise type based on the total hours spent by all members. ('C': Cardio, 'S': Strength Training, 'Y': Yoga).
- **`extraWorkout:`** takes the members array and the number of members as parameters, and finds and displays the members who need extra workout time. The minimum limit for the exercises is 5 hours.
- **`highCalBurners:`** takes the members array, the number of members, and the calorie limit as parameters. It finds and displays members who burned more than the specified calorie limit.
- **`calAvgWorkout:`** takes the members array and the number of members as parameters, calculates and returns the average workout time.

Project Name: LG6_Q2
File Name: Q2.cpp

`Fitness.txt`

```
15
101 JohnDoe C 6 500
102 AliceSmith S 4 350
103 MarkJohnson Y 5 200
104 EmilyBrown C 7 450
105 DavidWhite S 3 300
106 SarahMiller Y 8 220
107 JamesDavis C 10 480
108 LauraWilson S 2 320
109 RobertAnderson Y 6 210
110 OliviaMoore C 4 460
111 CharlesTaylor S 5 340
112 SophiaThomas Y 7 230
113 WilliamHarris C 9 490
114 EmmaMartin S 6 360
115 DanielThompson Y 3 190
```

Example Run:

```
FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 9
Enter your choice: -5
Enter your choice: 1

Member Fitness Data:
ID   Name                Type   Hours  Calories Burned
-----------------------------------------------------
101      JohnDoe           C      6    3000.00
102      AliceSmith        S      4    1400.00
103      MarkJohnson       Y      5    1000.00
104      EmilyBrown        C      7    3150.00
105      DavidWhite        S      3     900.00
106      SarahMiller       Y      8    1760.00
107      JamesDavis        C     10    4800.00
108      LauraWilson       S      2     640.00
109      RobertAnderson    Y      6    1260.00
110      OliviaMoore       C      4    1840.00
111      CharlesTaylor     S      5    1700.00
112      SophiaThomas      Y      7    1610.00
113      WilliamHarris     C      9    4410.00
114      EmmaMartin        S      6    2160.00
115      DanielThompson    Y      3     570.00

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 2

Enter member_t ID to search: 999

Member ID 999 not found.


FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 2

Enter member_t ID to search: 105

Member ID: 105
Type: S
Hours: 3
Calories Burned: 900.00

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 3

Enter calorie burn Limit to filter members: 2500

Members burning more than 2500.00 calories:
101-JohnDoe burned 3000.00 calories
104-EmilyBrown burned 3150.00 calories
107-JamesDavis burned 4800.00 calories
113-WilliamHarris burned 4410.00 calories

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 4

Most Active: 107-JamesDavis with 10 hours of exercise.

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 5

Most Popular Exercise: C with 36 total hours.

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 6

Average Weekly Workout Time: 5.67 hours

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 7

Members needing more workout (less than 5 hours):
102-AliceSmith needs 1 more hours
105-DavidWhite needs 2 more hours
108-LauraWilson needs 3 more hours
110-OliviaMoore needs 1 more hours
115-DanielThompson needs 2 more hours

FITNESS CENTER MENU
-------------------
1. Display All Members
2. Search a Member by ID
3. Display the high-calorie burners
4. Find the most active member
5. Find the most popular exercise type
6. Display Average Weekly Workout Time
7. Find Members Who Need Extra Workout Time
8. EXIT
Enter your choice: 8
```

Q3. A University opens a new graduate program and accepted students can be candidates for scholarships depending on some criteria. Accepted students' information (Student **name**, **cgpa**, **exam grades** consisting of **ales** grade and **yds** grade) is kept in the "**graduateStudents.txt**" file.

Write a C program that will read all of the information from the file into a **dynamically** created structure array. (The first line of the file consists of the number of students in the graduate program) The structure will also keep the **overall** grade and **scholarship percentage**. Then, display the number of students and the student information including the overall grade and the scholarship percentage as in the example run.

Write the following functions;

- **`calculateScholarship:`** takes a student as a parameter and calculates the overall grade and the scholarship percentage of the student.

- **`readFromFile:`** takes the file pointer, student array and the number of students as parameters, reads the students' information from the file into the array. The function also calculates the overall grade and the scholarship percentage using the function **`calculateScholarship.`**

The overall grade is going to be calculated by the sum of **30% of cgpa**, **45% of ales grade,** and **25% of yds grade**.

Overall Grade	Scholarship Percentage %
>90	100
>80	75
>65	50
<=65	0

- **`displayReport:`** takes the student array and the number of students as parameters, displays the number of students and the student information.

Project Name: LG6_Q3
File Name: Q3.cpp

`graduateStudents.txt`

```
6
JonSnow 87 95 89
DaenerysTargaryen 45 63 23
RachelGreen 45 67 87
RossGeller 45 80 90
JakePeralta 34 56 98
MaxBlack 65 98 89
```

Example Run:

```
There are 6 students in the Graduate program

STUDENT NAME         CGPA    ALES     YDS       OVERALL   SCHOLARSHIP PERCENTAGE
************         ****    ****     ***       *******   **********************
JonSnow              87.00   95.00   89.00      91.10            100 %
DaenerysTargaryen    45.00   63.00   23.00      47.60              0 %
RachelGreen          45.00   67.00   87.00      65.40             50 %
RossGeller           45.00   80.00   90.00      72.00             50 %
JakePeralta          34.00   56.00   98.00      59.90              0 %
MaxBlack             65.00   98.00   89.00      85.85             75 %
```

Additional Question

A vending machine will be simulated; the product list of the machine is stored in the file named **"products.txt"** with the **name, unit price,** and **quantity**. The first line of the file consists of the number of products. Initially, the program reads all the products' information from the file into a dynamically created structure array and displays them on the screen. While reading the product name you should use %[^0-9].

Then the program gets the product number from the customer and if the product does not run out of the machine, the customer inserts the money, otherwise gives an error message. After the purchase is finished, the remaining money should be returned to the customer. If the inserted money is not enough to purchase the product, the program gives an error message. When the product is run out of, the quantity of the item should be displayed OUT in the menu as in the example run. When the user enters -1 for the product number, the program terminates.

Write the following functions;

- **`readFromFile :`** takes a file pointer, a structure array, and a size as parameters. It reads the text file into the structure array and returns the size.
- **`display :`** takes a structure array and a size as parameters. It displays the content of the structure array. If the quantity is 0, it should display the quantity as "OUT".

Project Name: LG6_AQ
File Name: AQ.cpp

products.txt

```
10
Hans Freitag 2.00 8
Balconi 3.80 2
Finn Crip 2.40 1
Kit Kat 0.20 5
Pretzels 0.45 6
Pepero 0.90 15
Wonka 1.25 4
Hershey Cookie 0.75 3
Kettle Corn 0.35 1
Welch's Shacks 0.65 10
```

Example Run:

```
     PRODUCT       UNIT PRICE QUANTITY
  1. Hans Freitag     2.00         8
  2. Balconi          3.80         2
  3. Finn Crip        2.40         1
  4. Kit Kat          0.20         5
  5. Pretzels         0.45         6
  6. Pepero           0.90        15
  7. Wonka            1.25         4
  8. Hershey Cookie   0.75         3
  9. Kettle Corn      0.35         1
 10. Welch's Shacks   0.65        10

Enter the product number to purchase (to stop -1):9
Insert the money: 1.40
1.05 TL returned back

     PRODUCT       UNIT PRICE QUANTITY
  1. Hans Freitag     2.00         8
  2. Balconi          3.80         2
  3. Finn Crip        2.40         1
  4. Kit Kat          0.20         5
  5. Pretzels         0.45         6
  6. Pepero           0.90        15
  7. Wonka            1.25         4
  8. Hershey Cookie   0.75         3
  9. Kettle Corn      0.35       OUT
 10. Welch's Shacks   0.65        10

Enter product number to purchase (to stop -1):9
THERE IS NO MORE Kettle Corn

     PRODUCT       UNIT PRICE QUANTITY
  1. Hans Freitag     2.00         8
  2. Balconi          3.80         2
  3. Finn Crip        2.40         1
  4. Kit Kat          0.20         5
  5. Pretzels         0.45         6
  6. Pepero           0.90        15
  7. Wonka            1.25         4
  8. Hershey Cookie   0.75         3
  9. Kettle Corn      0.35       OUT
 10. Welch's Shacks   0.65        10

Enter product number to purchase (to stop -1):2
Insert the money: 2.50

Your money is not enough, and returned back
```

```
     PRODUCT       UNIT PRICE QUANTITY
  1. Hans Freitag     2.00         8
  2. Balconi          3.80         2
  3. Finn Crip        2.40         1
  4. Kit Kat          0.20         5
  5. Pretzels         0.45         6
  6. Pepero           0.90        15
  7. Wonka            1.25         4
  8. Hershey Cookie   0.75         3
  9. Kettle Corn      0.35       OUT
 10. Welch's Shacks   0.65        10

Enter product number to purchase (to stop -1):5
Insert the money: 1
0.55 TL returned back

     PRODUCT       UNIT PRICE QUANTITY
  1. Hans Freitag     2.00         8
  2. Balconi          3.80         2
  3. Finn Crip        2.40         1
  4. Kit Kat          0.20         5
  5. Pretzels         0.45         5
  6. Pepero           0.90        15
  7. Wonka            1.25         4
  8. Hershey Cookie   0.75         3
  9. Kettle Corn      0.35       OUT
 10. Welch's Shacks   0.65        10

Enter product number to purchase (to stop -1):-1
************************
Today 2 products sold
Total payment is: 0.80 TL
```