Department of Information Systems and Technologies

CTIS151 – Introduction to Programming

SPRING 2023- 2024

Lab Guide 13 – Week 11-1

OBJECTIVES : File Operations

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

Q1.

a) Write a C program that reads **10** numbers from a text file named **"numbers.txt"**. Then, it displays these values on the screen and calculate their multiplication.

numbers.txt

```
3 5 2 7 1 2 6 3 2 1 6 7 4 2 5 3 1 1
```

Project Name: LG13_Q1a
File Name: Q1a.cpp

Example Run:

```
Numbers read from the file are: 3 5 2 7 1 2 6 3 2 1
The multiplication of these numbers 15120
```

b) Modify the program **Q1a** that the file is, now, read until the **end of file**, instead reading a certain amount of numbers.

Example Run:

```
Numbers read from the file were: 3 5 2 7 1 2 6 3 2 1 6 7 4 2 5 3 1 1
The multiplication of these numbers were 76204800
```

Project Name: LG13_Q1b
File Name: Q1b.cpp

c) Modify the program **Q1b** that reads the numbers from the file **"numbers.txt"**. Then, writes the numbers which are even to the **even.txt**, writes other numbers to the file named **odd.txt**.

even.txt

```
2 2 6 2 6 4 2
```

odd.txt

```
3 5 7 1 3 1 7 5 3 1 1
```

Project Name: LG13_Q1c
File Name: Q1c.cpp

Q2.

Write a C program that reads a list of integers from a text file named **"integers.txt"** and displays the perfect numbers.

Write the following function;

- **`isPerfect()`** that gets an integer number then decides whether an integer is perfect or not. (A perfect number is a number whose sum of it is divisors equal to number itself excluding number itself and 1.)

integers.txt

```
6 10 38 91 65 99 32 67 80 91 28 15 18 23 37 45 86 72 41 62 38 93 27 68 90 12 25 80 33 57
```

Example Run:

```
Perfect numbers are :  6 28
```

Project Name: LG13_Q2
File Name: Q2.cpp

Q3.

Write a C program that reads a sentence from a file, **“sentence.txt”**, then writes all words with their lengths into a file, **output.txt**, and displays how many words are counted in each line of sentence.txt as in the example run.

sentence.txt

```
FAQ Frequently Asked Questions
PS Post Script
ROFL Rolling on the Floor Laughing
POV Point of View
ASAP As soon as possible
TBC To Be Continued
```

Example Run:

```
1. line --> 4 words
2. line --> 3 words
3. line --> 6 words
4. line --> 4 words
5. line --> 5 words
6. line --> 4 words
```

output.txt

```
FAQ 3
Frequently 10
Asked 5
Questions 9
PS 2
Post 4
Script 6
ROFL 4
Rolling 7
on 2
the 3
Floor 5
Laughing 8
POV 3
Point 5
of 2
View 4
ASAP 4
As 2
soon 4
as 2
possible 8
TBC 3
To 2
Be 2
Continued 9
```

Project Name: LG13_Q3
File Name: Q3.cpp

ADDITIONAL QUESTIONS

AQ1. Write a program that gets some arithmetic expressions, until # is entered. The program should write all the expressions to an output file named **expression.txt** and write near each expression “Balanced” if the pharanteses are opened and closed respectively or write “Unbalanced” if the pharanteses are not opened and closed respectively. And also give a message as “opening parantheses missing” or “closing parantheses missing” beside each expression.

Example Run:

```
Enter expressions # to finish:
((1+23)*12+(34-9)
(23+26*(10+(7*9)-12))
45+(82/2)-(45+2))
#
```

Project Name: LG13_AQ1
File Name: AQ1.cpp

expression.txt will contain:

```
((1+23)*12+(34-9)   UNBALANCED closing paranthese(s) is/are missing
(23+26*(10+(7*9)-12))   BALANCED
45+(82/2)-(45+2))   UNBALANCED opening paranthese(s) is/are missing
```

AQ2. Write a C program that reads numbers from a file **“nums.txt”,** finds the four digit numbers and writes them and their separated digits into a new file **“fourdigit.txt”**. The program also checks which of the four digit numbers provides the equation; **abcd = $(ab + cd)^2$** and displays these numbers as in the example run and writes them into a new file **“fourdigit.txt”**, and puts an asterisk (*) next to such numbers in the file **“fourdigit.txt”**.

Write the following functions;

- **isFourDigit** that gets an integer number and checks whether it is a four digit number or not.
- **separate** that gets a four digit integer number and separates the first two digits from the last two digits and returns them separately. For example; if the number is **abcd**, it returns **ab** and **cd**.

nums.txt

```
55
23698
1236
1233
101203
933
8260
874
3025
4728
5564
3
1987
```

fourdigit.txt

```
1236 12 36
1233 12 33
8260 82 60
3025 30 25 *
4728 47 28
5564 55 64
1987 19 87
```

Example Run:

```
Numbers providing the equation are
----------------------------------
3025
```

Project Name: LG13_AQ2
File Name: AQ2.cpp