

Lab Guide #10 – Week 8 – 1

OBJECTIVES	: Void functions with no parameters, Void functions with parameters, Functions that return a value
Instructors	: Serpil TIN
Assistants	: Berk ÖNDER, Efe Mert ŞAHINKOÇ, Hatice Zehra YILMAZ

Q1.

- a. Write a function named **displine** that displays a line containing 10 stars on the screen.
Write a C program that displays a single line on the screen using the function.

Project Name: LG10_Q1a

File Name: Q1a.cpp

Example Run:

- b. Modify the **Q1a.cpp** as follows:

Write the following functions;

- **menu**: that displays a menu on the screen (see in the example run).
- **dispRectangle**: that displays a 4x10 rectangle with the stars (*) using the function dispLine.
- **dispParallelogram**: that displays a 4x10 parallelogram with the stars (*) using the function dispLine.

Write a C program that displays a menu, reads and validates the choice, and displays a single line, a rectangle, or a parallelogram according to the user's choice.

Project Name: LG10_Q1b

File Name: Q1b.cpp

Example Run:

MENU

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT

Enter your choice: 9
Enter your choice: -1
Enter your choice: 1

MENU

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT

Enter your choice: 2

MENU

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT

Enter your choice: 3

MENU

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT

Enter your choice: 4

- c. Modify the **Q1b.cpp** as follows:

- **dispLine**: takes the symbol and the n-number of symbols to be displayed as parameters and displays the given symbol n times on a line.
- **dispRectangle**: takes the symbol and side lengths of the rectangle and displays the rectangle with the given symbols.
- **dispParallelogram**: takes the symbol and side lengths of the parallelogram and displays the parallelogram with the given symbols.

Project Name: LG10_Q1c

File Name: Q1c.cpp

Example Run:

```

MENU
1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT
Enter your choice: 1

Enter a symbol: &
Enter the number of symbols to be
displayed: 12
&&&&&&&&&&&&&&&&

```

```

MENU
1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT
Enter your choice: 2

```

```

Enter a symbol: $
Enter the sidel: 5
Enter the side2: 8
$$$$$$$$
$$$$$$$$
$$$$$$$$
$$$$$$$$
$$$$$$$$

```

MENU

```

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT
Enter your choice: 8
Enter your choice: 5
Enter your choice: 3

```

```

Enter a symbol: *
Enter the sidel: 4
Enter the side2: 9
*****
*****
*****
*****

```

MENU

```

1. Draw a single line
2. Draw a rectangle
3. Draw a parallelogram
4. EXIT
Enter your choice: 4

```

Q2. Write a C program that gets a positive integer from the user until a non-positive integer is entered and displays whether the numbers are prime or not.

Write the following function;

- **isPrime:** that gets an integer number as a parameter, and returns whether the number is a prime number or not.

NOTE: Prime numbers are natural numbers that are divisible by only 1 and the number itself.

Project Name: LG10_Q2

File Name: Q2.cpp

Example Run:

```

Enter a number: 17
17 is a prime number.

Enter a number: 6
6 is not a prime number.

Enter a number: 23
23 is a prime number.

Enter a number: 1
1 is not a prime number.

Enter a number: -2
Goodbye!

```

Q3. Write a C program that takes a positive integer number. The program sums up the digits of the integer to reduce it to a new integer, repeating the process until only a single digit remains.

E.g.: if the user enters 8974; 8974 is reduced to 28, which is reduced to 10, which is reduced to 1, providing the final value for termination.

Write the following function;

- **findSumOfDigits:** that takes an integer number as a parameter, finds and returns the sum of its digits.

Project Name: LG10_Q3

File Name: Q3.cpp

Example Run#1:

```

Enter a positive integer: -5
Enter a positive integer: 0
Enter a positive integer: 36987
Sum of digits = 33
Sum of digits = 6

```

Example Run#2:

```

Enter a positive integer: 698745
Sum of digits = 39
Sum of digits = 12
Sum of digits = 3

```

Q4. Write a C program to input base and the number, display the decimal equivalent using the function below.

Write the following function;

- **convertToBase**: that converts a number in a base smaller than 10 to its decimal value. The function takes the base and the number as input parameter and returns its decimal.

Project Name: LG10_Q4

File Name: Q4.cpp

Example Run #1:

Enter the base: 2

Enter the number: 1011

Decimal value of the number 1011 in base 2 is 11.

Example Run #2:

Enter the base: 9

Enter the number: 362

Decimal value of the number 362 in base 9 is 299.

ADDITIONAL QUESTIONS

AQ1.

Modify the question **Q4.cpp**, so that the program performs base conversion operations on numbers. The program takes a number and converts the given number into the given base.

Modify the function **convertToBase**, so that the function;

- takes a number, the base in which this number is represented and the base into which this number should be converted as input parameters, converts the given number into the given base, and returns the new number.

Note: The bases for given numbers must be between [2 – 10] while the bases into which they will be converted must be between [2 – 9]. If an invalid initial base is entered, the program must exit.

Example: $(210)_8 = (\dots)_5$

How to find the digits of a number(e.g. conversion of 210 in base 8 to base 10):

$$\begin{array}{r|l} 210 & 10 \\ 20 & 21 \\ \hline 10 & 20 \\ 10 & 10 \\ \hline 0 & 10 \end{array}$$

210 in base 8 is converted to base 10 as follows:

$$2 \times 8^2 + 1 \times 8^1 + 0 \times 8^0 = 128 + 8 + 0 = \underline{\underline{136}}$$

136 in base 10 is converted to base 5 as follows:

$$1 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 = 1000 + 0 + 20 + 1 = \underline{\underline{1021}}$$

Project Name: LG10_AQ1

File Name: AQ1.cpp

Example Run:

Enter the base and the number : 8 210

Enter the base to be converted : 5

Number in base 5 = 1021

Enter the base and the number : 2 1001001

Enter the base to be converted : 10

Enter the base to be converted : 1

Enter the base to be converted : 7

Number in base 7 = 133

Enter the base and the number : -2 0

AQ2.

Write a C program that reads several integer numbers from the user until a negative integer is entered and displays whether the given numbers are **automorphic** numbers.

*An **automorphic** number is a number whose square ends in the same digits as the number itself. For example, if a number n is automorphic, then n^2 will end in the digits of n (e.g., 76 because $76^2=5776$).*

Write the following functions;

- **findNumOfDigits**: takes an integer number as a parameter, finds and returns the number of digits in the given number.
- **isAutomorphic**: takes an integer number as a parameter, and checks if the number is automorphic or not. If the number is automorphic the function returns **1**, otherwise returns **0**.

Project Name: LG10_AQ2

File Name: AQ2.cpp

Example Run:

```
Enter a number (negative to stop): 0
Square: 0, last 0 digit(s): 0
0 is an Automorphic number.
```

```
Enter a number (negative to stop): 5
Square: 25, last 1 digit(s): 5
5 is an Automorphic number.
```

```
Enter a number (negative to stop): 76
Square: 5776, last 2 digit(s): 76
76 is an Automorphic number.
```

```
Enter a number (negative to stop): 88
Square: 7744, last 2 digit(s): 44
88 is NOT an Automorphic number.
```

```
Enter a number (negative to stop): 9376
Square: 87909376, last 4 digit(s): 9376
9376 is an Automorphic number.
```

```
Enter a number (negative to stop): 250
Square: 62500, last 3 digit(s): 500
250 is NOT an Automorphic number.
```

```
Enter a number (negative to stop): 625
Square: 390625, last 3 digit(s): 625
625 is an Automorphic number.
```

```
Enter a number (negative to stop): -3
```