

Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics

CSC326 – Operating Systems
Dr. Abdallah Dabboussi

Lab Assignment 3
Due: Monday October 10, 2022

Problem 1:

Write a C program that scans a string from the user and returns the number of letters, integers and special characters in the string.

Sample input/output:

Please enter a string:

HeLLoooCSC245!!!:~

The number of letters is: 10

The number of uppercase letters is: 6

The number of lowercase letters is: 4

The number of integers is: 3

The number of special characters is: 11

- *Submit your solution in a file called "Problem1.c".*

Problem 2:

Caesar Cipher is a simple encryption technique that replaces a letter by another letter based on the key. For instance, if the key k is 3 then the character is replaced by another character that is 3 positions after it, e.g. "a" is replaced by "d", "b" is replaced by "e", "c" is replaced by "f" and so on.

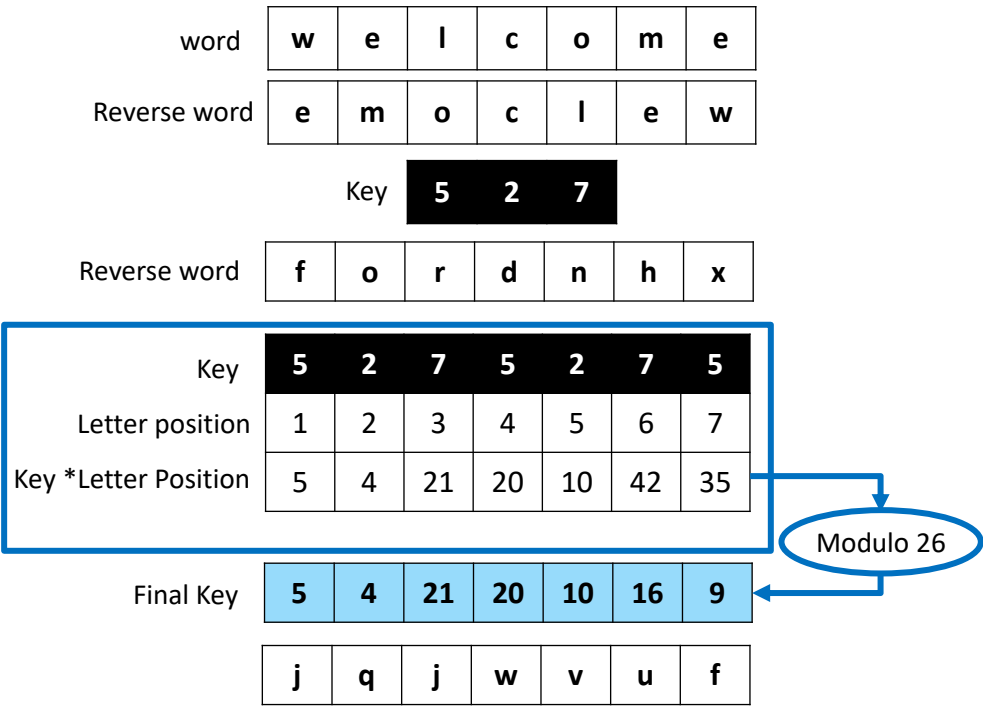
word : abcdefghijklmnopqrstuvwxyz

Key: 3

Cipher: defghijklmnopqrstuvwxyzabc

In this problem, we are interested in a "special Reverse" Caesar Cipher that will take an array of keys instead of one to encrypt a string as shown in the Figure below. Our encryption technique will consider the following:

- The word entered by the user will be ciphered in reverse order, e.g.: if the word to be ciphered is "welcome", then you start with the letter "e", then letter "m" and so on.
- The final key is generated based on the input array key as follows:
 1. First, the key will be repeated if its length is smaller than the word length
 2. The key will be then multiplied by the letter position, e.g: considering the first letter "e" (position=1), the key will be $5*1$, the second letter "m" located as position 2 will have a key of $2*2$, and so on...
 3. Make sure the final key value ranges between 0 and 25, by using modulo 26
- Use the final key to encrypt the letter by shifting k positions



Write a C program that:

1. Scans a string "word" from the user (less than 50 characters)
2. Prompts the user to enter N the size of the key (less than 20)
3. Prompts the user the enter N integers and store them in the key array "key"
4. Creates a new string "cipher" and ciphers the word based on the key as previously described and as shown in the sample input/output below.

Hint: if the ciphered character is greater than 'z' or less than 0, you need to set the cipher[i] to cipher[i] - 'z' + 'a' - 1; e.g.: 'z' + 1 should be ciphered as 'a'.

- Submit your solution in a file called "Problem2.c".

```
(kali㉿kali)-[~/Desktop]
$ ./Problem
Please enter the text you would like to cipher:
welcome
Please enter N the number of integers in the list:
3
Please enter the 3 integers:
5
2
7
Letter: e      Key: 5      Cipher: j
Letter: m      Key: 4      Cipher: q
Letter: o      Key: 21     Cipher: j
Letter: c      Key: 20     Cipher: w
Letter: l      Key: 10     Cipher: v
Letter: e      Key: 16     Cipher: u
Letter: w      Key: 9      Cipher: f
The text is: welcome
The ciphered text is: jqjwvuf
```

```
(kali㉿kali)-[~/Desktop]
$ ./Problem
Please enter the text you would like to cipher:
hello
Please enter N the number of integers in the list:
2
Please enter the 2 integers:
13
8
Letter: o      Key: 13     Cipher: b
Letter: l      Key: 16     Cipher: b
Letter: l      Key: 13     Cipher: y
Letter: e      Key: 6      Cipher: k
Letter: h      Key: 13     Cipher: u
The text is: hello
The ciphered text is: bbyku
```

```
(kali㉿kali)-[~/Desktop]
$ ./Problem
Please enter the text you would like to cipher:
abcdefghijklmnopqrstuvwxyz
Please enter N the number of integers in the list:
1
Please enter the 1 integers:
3
Letter: z      Key: 3      Cipher: c
Letter: y      Key: 6      Cipher: e
Letter: x      Key: 9      Cipher: g
Letter: w      Key: 12     Cipher: i
Letter: v      Key: 15     Cipher: k
Letter: u      Key: 18     Cipher: m
Letter: t      Key: 21     Cipher: o
Letter: s      Key: 24     Cipher: q
Letter: r      Key: 1      Cipher: s
Letter: q      Key: 4      Cipher: u
Letter: p      Key: 7      Cipher: w
Letter: o      Key: 10     Cipher: y
Letter: n      Key: 13     Cipher: a
Letter: m      Key: 16     Cipher: c
Letter: l      Key: 19     Cipher: e
Letter: k      Key: 22     Cipher: g
Letter: j      Key: 25     Cipher: i
Letter: i      Key: 2      Cipher: k
Letter: h      Key: 5      Cipher: m
Letter: g      Key: 8      Cipher: o
Letter: f      Key: 11     Cipher: q
Letter: e      Key: 14     Cipher: s
Letter: d      Key: 17     Cipher: u
Letter: c      Key: 20     Cipher: w
Letter: b      Key: 23     Cipher: y
Letter: a      Key: 0      Cipher: a
The text is: abcdefghijklmnopqrstuvwxyz
The ciphered text is: cegikmoqsuwya
```

Problem 3:

In this problem, we are interested in the temperature. Create a "**Temperature**" struct that has two attributes: (1) *temp* which is a real number indicating the temperature value and (2) the *degree* which is a character representing the degree in Celsius 'C' or in Fahrenheit 'F'.

Write a program to implement and test your struct and the following methods:

- *ChangeTemp* method that takes a temperature value t , a degree character d and a temperature T by reference as input and changes the attributes of the temperature T based on the t and d . *Note: Make sure that the input character d is either 'F' or 'C', i.e.: if d is 'F', your method should set the degree to 'F', otherwise to 'C'*
- *getCelsius* method that takes as input a temperature T and returns the value of the temperature in Celsius. Your method should first check the degree. If the temperature degree is Celsius, it should return the value of the temperature. Otherwise, the temperature is in Fahrenheit, and your method should convert the temperature value to Celsius as follows: $C = (F - 32)/1.8$
- *getFahren* method that takes as input a temperature T and returns the value of the temperature in Fahrenheit. Your method should first check the temperature degree. If the temperature is in Fahrenheit, it should return the value of the temperature. Otherwise, the temperature is in Celsius, and your method should convert the temperature value to Fahrenheit as follows: $F = 1.8 \times C + 32$
- *Submit your solution in a file called "Problem3.c".*

Problem 4:

In this problem, we are interested in spheres represented by their center coordinates x , y and z and their *radius*. Create a "**Sphere**" struct that implements the following functions:

- *ScaleSphere* method that takes a positive integer a and a sphere by reference as input and scales the radius of the sphere by a , i.e.: $radius \times a$
- *ComputeArea* method that takes a Sphere as input and returns the outside surface area of the sphere expressed as follows: $4\pi(radius)^2$
- *ComputeVolume* method that takes a Sphere as input and returns the volume of the sphere expressed as follows: $\frac{4}{3}\pi(radius)^3$
- *withinSphere* method that checks if a point $P(a, b, c)$ is inside the sphere. Accordingly, the method *withinSphere* takes as input a Sphere and three real numbers a , b and c where a , b and c are the coordinates of a point

P . Your method should return *true* if the point P is inside the sphere; i.e.: the distance between the center of the sphere and P is less than the *radius*

- *MoveSphere* method that takes a sphere by reference as input, as well as three real numbers a , b and c . The method will modify the coordinated of the center of the sphere by moving it by (a, b, c) , hence, the new value of x will be $x + a$, the new value of y will be $y + b$, and the new value of z will be $z + c$.

Write a program to test your struct and functions as shown in the sample input/output below.

Please enter the coordinates of the center of the sphere:

1.5

2

1.5

Please enter the radius of the sphere: 10

Sphere 1: 0(x=1.50, y= 2.00, z= 1.50), radius = 10.00

Sphere 1 scaled by 2: 0(x=1.50, y= 2.00, z= 1.50), radius = 20.00

Sphere 1 area= 5024.00

Sphere 1 volume= 1256.00

Please enter the coordinates of the point P:

3

5

2

P(x=1256.00, y= 62.80, z= 0.00) is located inside the sphere

Move the sphere by (10,0,5):

Sphere 1: 0(x=11.50, y= 2.00, z= 6.50), radius = 20.00

Note: Don't forget to add " #include <stdbool.h>" to use "bool". You may need to use scanf("%lf", &a) to scan a real number (double) from the user. You can also assume $\pi = 3.14$.

- *Submit your solution in a file called "Problem4.c".*

Problem 5:

Write a C program that implements the following:

```
//Declare an integer a and set its value to 20
```

```
==> int a=20;
```

```
//Declare a pointer integer b and binds the integer reference a to b
```

```
==> int *b = &a;
```

```
//Declare an integer c and set its value to the value of a
```

```
==> int c=a;
```

```
Value of a = 20
```

```
Value of b = 6efc21d4
```

Value of *b = 20
Value of c = 6efc21d4
Value of &a = 6efc21d4 ==> Address of a
Value of b = 6efc21d4
Value of &b = 6efc21d4
Value of &c = 6efc21d0 ==> Address of c
//Modify the value of a and set it to 40
==> a=40;

What do you expect the updated values of a , $*b$ and c to be?

- *Submit your solution in a file called "Problem5.c".*