FACULTY OF COMPUTERS AND INFROMATION, CAIRO UNIVERSITY

CS112: Programming I
Year 2017-2018

Second Semester

Assignment 2 – Version 4.0

Course Instructors:

Dr. Mohammed Al-Ramly
Dr. Amin Allam

Revision History

Version 1.0	By Dr Mohammed El-Ramly 20 Feb. 2018	Main Doc
Version 2.0	By Dr Mohammed El-Ramly 22 Feb. 2018	Errors Fixed
Version 3.0	By Dr Mohammed El-Ramly 24 Feb. 2018	Elaborations in red
Version 4.0	By Dr Mohammed El-Ramly 28 Feb. 2018	Pb 0, 1 fixed

Cairo University, Faculty of Computers and Information

CS111: Fundamentals of CS
Assignment 2 (5 marks + 2 bonus) – Version 4.0

Objectives

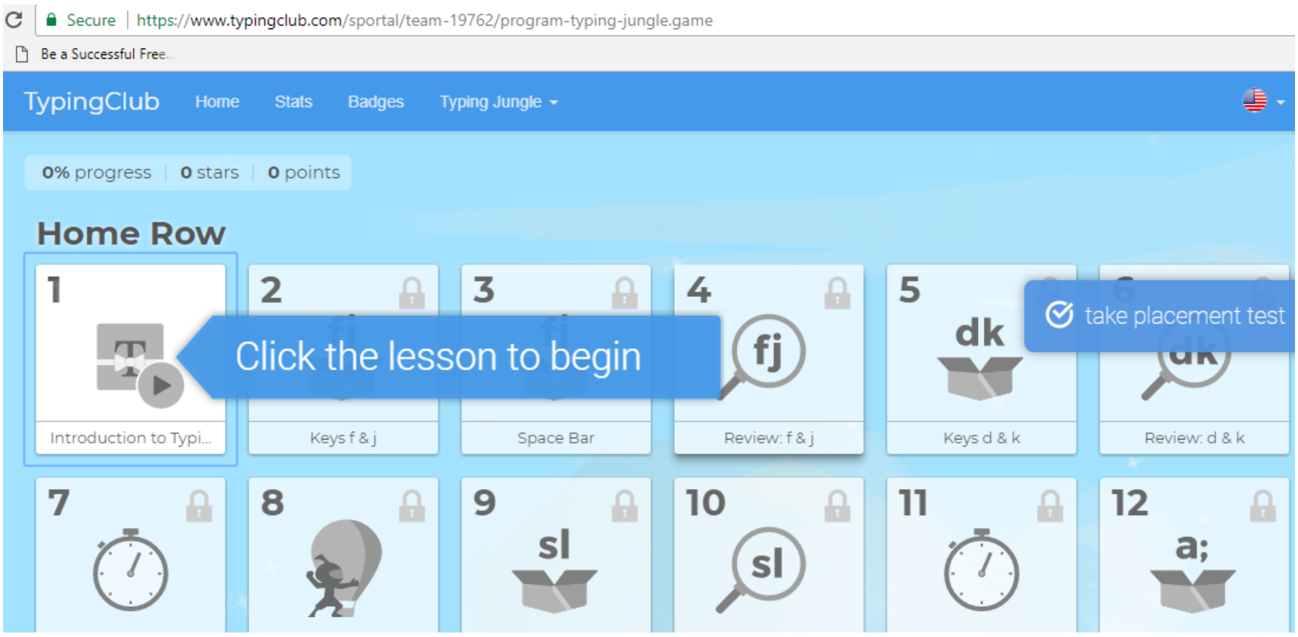
This assignment trains students on algorithms, flowcharts and C++ Basics.

Instructions

1. **It is very important to collect course work marks in order to pass easily and get a good grade. من المهم للغاية حسن أداء أعمال السنة لتنجح بسهولة و تحصل على تقدير مرتفع**
2. **These instructions must be followed to get the full marks. يجب اتباع هذه التعليمات بكل دقة**
3. **Deadline is Friday 2 March 2018 @ 11:59 pm.**
4. Weight of the assignment is **5 marks** + you can earn **2 bonus** marks.
5. Students will forms teams of three students **from the same group** whose IDs **do not end with the same digit.** For example, 2017023, 20170433 and 20170124 cannot be in one team because two of them have IDs ending with 3. الفريق من ٣ طلاب لا ينتهى رقم بطاقة الكلية لهم بنفس الرقم
6. Please submit **only work that you did yourself**. If you copy work from your friend or book or the net **you will fail the course**. تسليم حلول منقولة من أى مصدر يؤدى إلى الرسوب فى هذا المقرر لا تغش الحل أو تنقله من أى مصدر و تعالى و اسألنى فى أى شئ لا تفهمه

Task 1 (1 mark) – Individual Task

1. This is to be done by each individual student.
2. Create an account on https://www.typingclub.com and login.
3. Do the **first 100 lessons** in the screen below and practice for at least 3 hours.
4. Print screens that show your account name and statistics of how long you used the program.



Task 2 (1.5 mark) – Individual Task

Each team member will solve one of these problems, by designing (an algorithm or flowchart) and programming an encryption algorithm. *Encryption is changing a message so it cannot be understood.* The program implemented will depend on the student's last digit in his ID. For example if his ID number is 20170080, then he solves problem **0** and if his number is 20170236, he solves problem **6**. If the student solves the **wrong problem,** he gets **ZERO**. In each problem, the student develops a program that takes a message from the user and **encrypts it** using a **cipher algorithm.** The program should also allow the user to **decipher** a given message to get the original message. Each student will implement a different cipher. So, it has three options like this. It loops until the user exits. **You should test your program very well.**

```
Ahlan ya user ya habibi.
What do you like to do today?
1- Cipher a message
2- Decipher a message
3- End
>> 1
Please enter the message to cipher:
.....
```

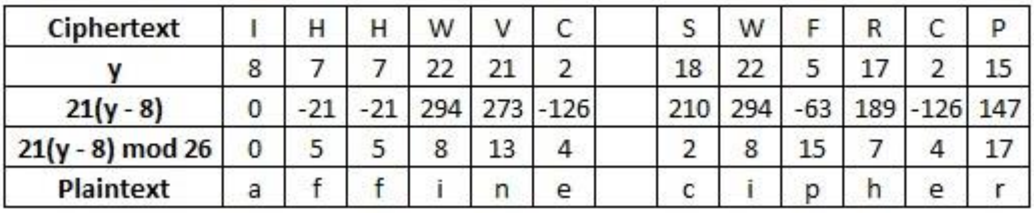
0. **Affine Cipher**

In affine cipher each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter. Letter A is given number 0 and letter Z is given number 26. Each letter is encrypted with the function $(5x + 8) \bmod 26$. The decryption function is $21(y - 8) \bmod 26$. See examples at: https://cryptii.com/affine-cipher/.

Example

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext	A	F	F	I	N	E	C	I	P	H	E	R
x	0	5	5	8	13	4	2	8	15	7	4	17
$(5x + 8)$	8	33	33	48	73	28	18	48	83	43	28	93
$(5x + 8) \bmod 26$	8	7	7	22	21	2	18	22	5	17	2	15
Ciphertext	I	H	H	W	V	C	S	W	F	R	C	P



Ciphertext	I	H	H	W	V	C		S	W	F	R	C	P
y	8	7	7	22	21	2		18	22	5	17	2	15
21(y - 8)	0	-21	-21	294	273	-126		210	294	-63	189	-126	147
21(y - 8) mod 26	0	5	5	8	13	4		2	8	15	7	4	17
Plaintext	a	f	f	i	n	e		c	i	p	h	e	r

Make a general version that takes three parameters *a*, *b* and *c* and does the encryption and decryption according to these equations:

E(x) = (a x + b) mod 26   where x is the letter to cipher.

D(y) = c (y - b) mod 26   where y is the letter to decipher.

a, b, c are arbitrary positive integers that satisfy the condition **(a * c) mod 26 = 1**

CS111: Fundamentals of CS
Assignment 2 (5 marks + 2 bonus) – Version 4.0

Cairo University, Faculty of Computers and Information

1. Caesar Cipher

Caesar cipher is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter at some fixed number of positions down the alphabet. For example, with a **right** shift of 3, A would be replaced by D, B would become E, X becomes A, and so on. **User should enter the number of shifts he wants.**

Example (assume a **right** shift by 3)

```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVW
```

2. Atbash Cipher

The Atbash cipher is a very common, simple cipher. Basically, when encoded, an "A" becomes a "Z", "B" turns into "Y", etc. See http://rumkin.com/tools/cipher/atbash.php

Example

```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   ZYXWVUTSRQPONMLKJIHGFEDCBA

Plain:    MOHAMMAD ELRAMLY
Cipher:   NLSZNNZW  VOIZNOB
```

3. ROT13 Cipher

It is a simple letter substitution cipher that replaces a letter with the 13th letter after it, in the alphabet. Because there are 26 letters (2×13) in the basic Latin alphabet, ROT13 is its own inverse; that is, to undo ROT13, the same algorithm is applied, so the same action can be used for encoding and decoding. See http://www.rot13.com/

4. Baconian Cipher

To encode a message, each letter of the plaintext is replaced by a group of five of the letters 'A' or 'B'. This replacement is a binary encoding. For example, SAMY will be baaba aaaaa abbaa bbaaa.

Letter	Code	Binary	Letter	Code	Binary	Letter	Code	Binary	Letter	Code	Binary
A	aaaaa	00000	G	Aabba	00110	N	abbab	01101	U	babaa	10100
B	aaaab	00001	H	Aabbb	00111	O	abbba	01110	V	babab	10101
C	aaaba	00010	I	Abaaa	01000	P	abbbb	01111	W	babba	10110
D	aaabb	00011	J	Abaab	01001	Q	baaaa	10000	X	babbb	10111
E	aabaa	00100	K	Ababa	01010	R	baaab	10001	Y	bbaaa	11000
F	aabab	00101	L	Ababb	01011	S	baaba	10010	Z	bbaab	11001
			M	Abbaa	01100	T	baabb	10011			

See http://rumkin.com/tools/cipher/baconian.php

5. Simple Substitution Cipher.

In this cipher, a replacement alphabet is used to replace each letter by another one. See http://practicalcryptography.com/ciphers/simple-substitution-cipher/
For example, if we use this cipher alphabet:

```
plain alphabet : abcdefghijklmnopqrstuvwxyz
cipher alphabet: phqgiumeaylnofdxjkrcvstzwb
```

The, we can encrypt the following sentence as follows:

```
Plain text : I love C plus plus
Cipher text: a ndsi q xnvr xnvr
```

Create a general version that builds the cipher alphabet using a **given key of 5 unique letters**. The user enters the key to cipher a message and the same key to decipher the message. The cipher alphabet is built by adding the remaining 21 letters in order after the key letters. For example, if the user enters "**zebra**" as the key, then:

```
plain alphabet : a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher alphabet: z e b r a c d f g h i j k l m n o p q s t u v w x y
```

The, we can encrypt the following sentence as follows:

```
Plain text : I love C plus plus
Cipher text: g jmua b njtq njtq
```

If the user enters "cairo" as the key, then:

```
plain alphabet : a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher alphabet: c a i r o b d e f g h j k l m n p q s t u v w x y z
```

The, we can encrypt the following sentence as follows:

```
Plain text : I love C plus plus
Cipher text: f jmvo i njus njus
```

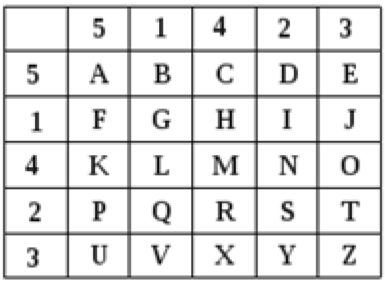
6. Polybius Square Cipher

See how it works at https://www.wikihow.com/Use-a-Polybius-Square using numbers. Each letter is replaced with two numbers according to the following table. Write another version where top and left numbers are given as a key, e.g., instead of 1 | 2 | 3 | 4 | 5 , user can enter the key 2 | 4 | 5 | 3 | 1 but you will need to have the same key to decipher the message. So, if the sender enters the key 1 2 3 4 5

[redacted]

```
Plain text : I  love      C  plus     plus
Cipher text: 24 32355215 13 41325144 41325144
```

If the user enters the key 5 1 4 2 3, then we have



	5	1	4	2	3
5	A	B	C	D	E
1	F	G	H	I	J
4	K	L	M	N	O
2	P	Q	R	S	T
3	U	V	X	Y	Z

```
Plain text : I  love      C  plus      plus
Cipher text: 12 41433153 54 25413522 25413522
```

To decipher the message, you must enter the same key used to encrypt it. So, the sender who ciphers the message and the receiver who decipher it BOTH must know the same key.

7. Morse Code

It is a code consisting of two symbols **dot** and **dash** and used to in the telegraph system in the past and also communicate messages in primitive ways. See http://www.unit-conversion.info/texttools/morse-code/

Develop a program to translate a message to Morse code and the opposite. Assume that each letter is separated by one space from the next and that each word is separated by three spaces from the next.

Example:

```
Plain text: I love C plus plus
Morse text:
..   .-.. --- ...- .   -.-.   .--. .-.. ..- ...   .--. .-.. ..- ...
```

8. XOR Cipher

In this cipher, a secret key consisting of one letter is give. Then each letter of the message goes through XOR operation with the **secret letter**. The output is printed in **text** and **hexadecimal**. The original message can be recovered from the encrypted message by the same algorithm, XOR with the secret letter.

Example:

```
Secret key is 'P'= 01010000
Plain text:  ILIKECPLUSPLUS
Cipher text: y}yzus`|eb`}ec  (Hexa 797d797a7573607c6562607d6563)
```

See http://md5decrypt.net/en/Xor/#results for trying the tool and use this to convert hexa to text https://www.rapidtables.com/convert/number/hex-to-ascii.html

Some letters combined with some keys will produce unreadable characters. **In reality, this is not a problem.** But for us, we like to be able to reenter the ciphered message. **So, you need to print the ciphered text in hexa also (**as above) and allow the user to **enter the message to decipher as hexa.**

9. Rail-fence Cipher

See the details at http://practicalcryptography.com/ciphers/classical-era/rail-fence/.

CS111: Fundamentals of CS
Assignment 2 (5 marks + 2 bonus) – Version 4.0

Task 3 (1 mark) – Individual Task

Each team member will solve one of these problems, by designing (algorithm or flowchart) and programming it to solve one of these problems. The program implemented will depend on the student's last digit in his ID. For example if his ID number is 20170080, then he solves problem **0** and if his number is 20170236, he solves problem **6**. If the student solves the **wrong program,** he gets **ZERO**. Problems are from "Problem solving with C++" by Walter Savitch, 9th ed.

0. Solve problems (1) Chap 2, page 106, problem 2
1. Solve problems (1) Chap 2, page 107, problem 6
2. Solve problems (1) Chap 2, page 107, problem 7
3. Solve problems (1) Chap 2, page 108, problem 11
4. Solve problems (1) Chap 2, page 109, problem 13
5. Solve problems (1) Chap 2, page 109, problem 14
6. Solve problems (1) Chap 3, page 173, problem 5
7. Solve problems (1) Chap 3, page 173, problem 6
8. Solve problems (1) Chap 3, page 173, problem 7
9. Write a program that computes the square root using Bakhshali method described in the lecture.

Task 4 (1.5 mark) – Team Task

1- The team will cooperate together and test the code of each other.
2- Team should fully understand the code of each other.
3- Team should ensure all members submitted code that follows the standard below and is organized. Code must have good style and comments in each file.
4- Team should solve one of the following problems according to the group they are in. All the problems are taken from the book "Problem solving with C++" by Walter Savitch, 9th ed.

- **GE, G1, G9, G17** – Solve chap 3, page 175, problem 3
- **G2, G10, G18** – Solve chap 3, page 176, problem 7
- **G3, G11, G19** – Solve chap 3, page 178, problem 11
- **G4, G12, G20** – Solve chap 3, page 177, problem 8
- **G5, G13, G21** – Solve chap 3, page 176, problem 7
- **G6, G14, G22** – Solve chap 3, page 175, problem 3
- **G7, G15, G23** – Solve chap 3, page 174, problem 2
- **G8, G16, G24** – Solve chap 3, page 174, problem 1

Bonus1: Task 5 (2 mark)

To get 2 bonus marks, the team should implement a cipher program the implements **all the ten algorithms** in task 2. The program should allow the user to choose the specific cipher (or decipher) he wants to apply.

CS111: Fundamentals of CS
Assignment 2 (5 marks + 2 bonus) – Version 4.0

Cairo University, Faculty of Computers and Information

How to Prepare and Deliver the Solution

Submitting the Solution

1. **Team will submit into acadox the following:**
   - A zip file with a pdf document with their names and IDs and list of programs they made. Name the zip file
     CS112-2018-2nd –Group-YourID-YourID-A2.pdf
     CS112-2018-2nd –Group-YourID-YourID-A2.zip
   - The document should have a cover page similar to this one. (This is for another course)
   - The source code of the programs in separate folders.
   - An algorithm or flow chart of each program that explains it.
   - Screen shots of https://www.typingclub.com account.
2. **Team will submit in paper to the TA following:**
   - A printed version of the pdf document with (1) cover page (2) team names, IDs and list of programs done, (3) flowcharts or algorithms of each program and (4) screen shots of https://www.typingclub.com accounts.

Cairo University
Faculty of Computers and Information

CS251

Software Engineering I

Project Description

2015

Version 2.0

Project Team

Dr Mohammad El-Ramly

m.elramly@fci-cu.edu.eg

Coding Style

The program should follow proper coding style for C++ as shown below.

<table>
<tr><td colspan="2"><b>1- Variable names must be in mixed case starting with lower case.</b></td></tr>
<tr><td colspan="2"><code>line, savingsAccount</code></td></tr>
<tr><td colspan="2"><b>2. The prefix <i>n</i> should be used for variables representing a number of objects.</b></td></tr>
<tr><td colspan="2"><code>nPoints, nLines</code></td></tr>
<tr><td colspan="2"><b>3. Iterator variables should be called i, j, k etc.</b></td></tr>
<tr><td colspan="2"><code>for (int i = 0; i < nTables); i++) { : }</code></td></tr>
<tr><td colspan="2"><b>4. The prefix <i>is</i> should be used for Boolean variables and methods.</b></td></tr>
<tr><td colspan="2"><code>isSet, isVisible, isFinished, isFound, isOpen</code></td></tr>
<tr><td colspan="2"><b>5. The conditional should be put on a separate line.</b></td></tr>
<tr><td colspan="2"><pre>if (isDone) // NOT: if (isDone) statement1;
   statement1;</pre></td></tr>
<tr><td colspan="2"><b>6. Block layout should be as illustrated in example 1 below (recommended) or example 2</b></td></tr>
<tr><td><pre>while (!done) {
   doSomething();
   done = moreToDo();
}</pre></td><td><pre>while (!done)
{
   doSomething();
   done = moreToDo();
}</pre></td></tr>
</table>

(Taken from http://geosoft.no/development/cppstyle.html)

CS111: Fundamentals of CS
Assignment 2 (5 marks + 2 bonus) – Version 4.0

Cairo University, Faculty of Computers and Information

Program Header

Each program should start with a header explaining what it is and who authored it. It should also have the date.

- لابد أن يحتوى كل برنامج على **تعليقات و إيضاحات كافية** و أن يبدأ بالتعليق التالى:

```
// FCI – Programming 1 – 2018 - Assignment 2
// Program Name:             xxxxxx.cpp
// Last Modification Date:  xx/xx/xxxx
// Author1 and ID and Group:     xxxxx xxxxx
// Author2 and ID and Group:     xxxxx xxxxx
// Author3 and ID and Group:     xxxxx xxxxx
// Teaching Assistant:           xxxxx xxxxx
// Purpose:..........
```

Academic Honesty Declaration

Each group should fill this form and submit with the report to the TA.

- لابد أن يملأ كل فريق هذا القسم و يقدمه مع التقرير للمعيد .

Faculty of Computers and Information **جامعة القاهرة – كلية الحاسبات و المعلومات**

Programming 1 – 2018- Assignment 2 **الفرقة الأولى – برمجة الحاسبات ١ – ٢٠١٨ - المسألة ٢**

Group ....المجموعة Date .... التاريخ Name...اسم الطالب

Group ....المجموعة Date .... التاريخ Name...اسم الطالب

Group ....المجموعة Date .... التاريخ Name...اسم الطالب

We give oath that we have fully authored all the programs we submitted for Assignment 2 and we did not copy work from the net, from other colleagues or from any sources.

نقسم بالله العظيم نحن الموقعون أدناه أننا قد قمنا بتنفيذ هذه المسألة Assignment 2 بأنفسنا و لم نغش مطلقا أو ننقل جهد غيرنا للحصول على درجات بغير حق أو نعطى مجهودنا للآخرين بغير حق و الله على ما نقول شهيد (من يتحرج من صيغة القسم لسبب دينى يكتب ما يناسب معتقده)

Signature .... التوقيع Signature .... التوقيع

Signature .... التوقيع

Marking Criterion

1 mark	For evidence of using typingclub.com and doing 30 lessons and 3 hours
1.5 marks	For correct implementation of encryption and decryption algorithms
1 mark	For correct implantation of the assigned problem
1 mark	For correct implementation of the group problem.
0.5 marks	For group collaboration and following coding style.