ASSIGNMENT COVERSHEET

UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER & NAME	NAME OF STUDENT(s) (PRINT CLEARLY)	STUDENT ID(s)
48430 Fundamentals of C Programming	Pin-Hsun Huang (Charlie) Jose Landi	14247845 14172530

STUDENT EMAIL	STUDENT CONTACT NUMBER
Pin-Hsun.Huang@student.uts.edu.au jose.l.landi@student.uts.edu.au	0432562636

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Robin Aldridge-Sutton	05	05/11/2022

ASSESSMENT ITEM NUMBER & TITLE

Assessment task 3: Group Project

☒ I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.
☒ I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements.
☒ I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

Statement of collaboration:

Signature of student(s) PH **Date** 05/11/2022

✂----------

ASSIGNMENT RECEIPT

To be completed by the student if a receipt is required

SUBJECT NUMBER & NAME	NAME OF TUTOR

SIGNATURE OF TUTOR	RECEIVED DATE

Table of Contents

Assignment Task 3 Report

Design Project – Caesar Cipher Program for a company to store their employee's information in a secured way through the use of encryption/decryption

Work Split

Charlie Huang: Encryption Function + Report

Jose Landi: Decryption Function + Report

David Kurniawan: Cannot contact this group member (tried both MS teams and email)

Matthew Lee: Cannot contact this group member (tried both MS teams and email)

Background

Our group collaborate in teams to create a software of a professional calibre that manipulates actual data as part of this research-inspired project. The ability to work in a team, manage a project, and communication will all be developed. Our group has chosen the topic of Caesar Cipher as our C project. It is a program that manipulates real-world data through encryption and/or decryption to store employees' details within a company.

Caesar Cipher is a simple encryption and decryption method developed a long time ago, around 100 BC. It was one of the earliest known ciphers developed by Julius Caesar to send secret messages to his generals in the field. If one of his messages got intercepted, his opponent could not read them. This could be very useful in the war time when you do not want your opponent to know the messages.

According to legend, Caesar (yes, that Caesar) once "encrypted" (i.e., concealed in a reversible manner) private messages by moving each letter a certain number of times. As an example, he could write A as B, B as C, C as D,..., and so on, winding up alphabetically with Z as A. Caesar might instead write IFMMP to say HELLO to someone. Such signals from Caesar would have to be "decrypted" by the recipients by shifting the letters in the opposite way by the equal number of places.

The confidentiality of this "cryptosystem" depended on the knowledge of a secret, the number of locations where Caesar had moved his letters, by only Caesar and the receivers (e.g., 1). By modern standards, not very secure, but hey, if you're possibly the first person in the world to do it, quite secure!

Plaintext generally refers to unencrypted text. Ciphertext is the usual name for encrypted text. A key is the name given to the used secret.

The process for encrypting HELLO using a key 1 that gives IFMMP is as follows:

plaintext	H	E	L	L	O
+ key	1	1	1	1	1
= ciphertext	I	F	M	M	P

In more technical terms, the cypher known as Caesar's algorithm encrypts communications by "rotating" each letter according to locations. To explain it more precisely, if is some plaintext (i.e., an unencrypted message), is the character in, and is a secret key (i.e., a non-negative integer), then each letter in the ciphertext, is computed as

$$c_i = (p_i + k)\%26$$

Problem requirement

Create a programme called Caesar that uses Caesar's cypher to encrypt and decrypt messages.

Implement a program in a file called main.c which has the function of taking in a string and an integer (key) as an input and return the encrypt representation of the string using the key. The program will also have the function to achieve the other way round, which is talking the encrypted message and decrypt into the original message. The program will first ask the user on whether they want to encrypt or decrypt a message. Based on the selection, different function will be used in order to fulfil the requirements.

If any of the characters of the command-line argument is not a decimal digit, your program should print the message from main a value of 1.

Keep in mind the key is not guaranteed to be less than or equal to 26. All integer non-negative values of key less than 23126 should be supported by your programme. So, if the user selects a value for key that is too large or nearly too large to fit in an int, you don't need to worry about your application breaking at any point. (Keep in mind that an int can overflow.) However, even if the key is more than 26, the alphabetical characters that were fed into your programme must stay in the output of your programme. If k = 27, for example, A should not become even though is 27 spots away from A in ASCII, according to asciichart.com; instead, A should become B as B is 27 locations away from A, provided you wrap around from Z to A. Below is an ASCII chart reference on the number corresponding to the ASCII characters that we are going to decode.

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

Decimal ASCII Chart

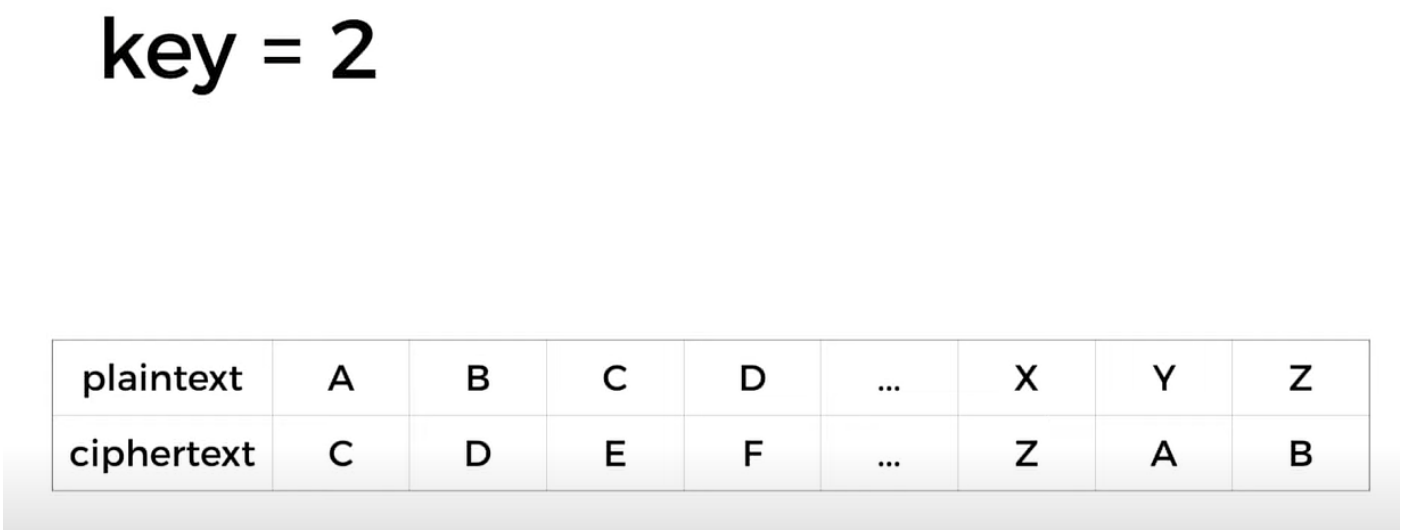
According to the user's selection of encode or decode, the software must output plaintext or ciphertext, with non-alphabetical characters remaining unmodified and each alphabetical character in the plaintext "rotated" by k places.

The programme must maintain case, thus even while capitalised letters are rotated, they must still be capitalised characters, and vice versa for lowercase letters.

The program must print a newline after ciphertext/original text output. Returning 0 from main should then cause the programme to terminate.

Problem analysis

In Caesar, the task is going to be encipher or encrypt the username of an employee by shifting all of the letters in that text by a certain amount, which will be the key entered by the user. What the program is ultimately going to do is first prompt the user whether they want to encrypt or decrypt a text, and then based on the selection, we are going to prompt the users for the key, which is the amount that the letters are going to be shifted by, then we are going to encipher that plain text by shifting the letters by that key, then finally print out the ciphertext.

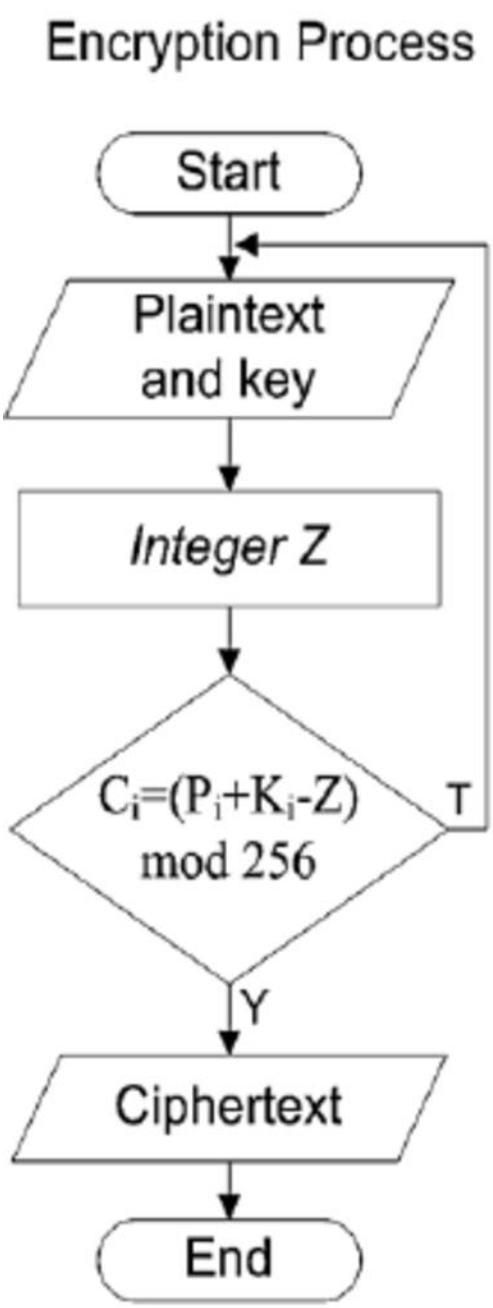


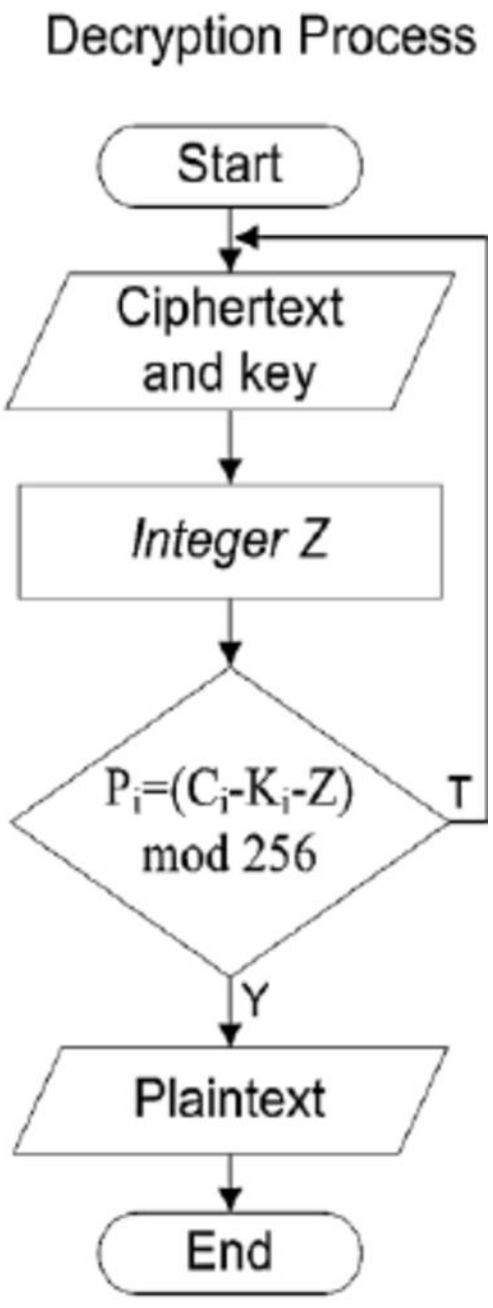
If the key is 2, as the above image, A will be shifted to C and B will be shifted to D, and so on. When we get to Y, if we shift Y by 2 characters, we would go pass the boundary of the alphabet. So what we will instead want to do is wrap around the beginning of the alphabet, so that Y becomes A, when we shift it by 2 and Z becomes B, when we shift it by 2

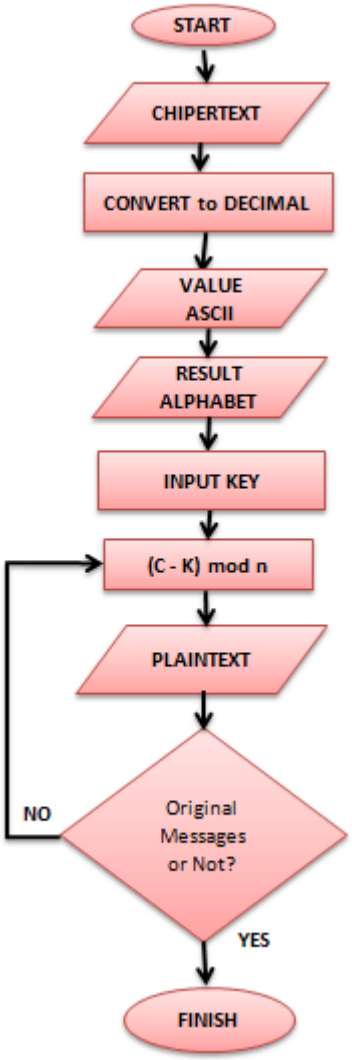
Design Solution (flowchart)

Our group decided to make a flowchart to convert our idea into a process that we can follow when we implement the coding part.

A flowchart is a diagram that shows a system, a process, or an algorithm for a computer. It includes a summary of the crucial steps to addressing the problem in addition to being a diagrammatic representation of the solution to a specific problem.







Implementation and test

Coding is completed in the implementation phase, and the software created serves as the input for the testing step that follows. The developed code is rigorously tested throughout the testing process to find any software flaws. When a fault is repaired, it is retested and entered the defect tracking tool.

The third stage of the software development cycle is the implementation of software solutions. The source code is written and tested at this stage of the development process. In other words, a programming language is used to apply the ideas and strategies developed in stage two.

Our group implemented our design by splitting it into smaller functions so everyone can do the coding part and work is done evenly.

For example, Charlie wrote the Encryption method which has no return type and takes an employee_t type struct as an input. The method whenever it's gets called will output the encrypted message. Basically turning a text into an encrypted text by applying the Caesar Cipher algorithm. E.g., input: hello with key = 5. Output: mjqqt.

```
void Caesar_Cipher_Encryption(employee_t e)
{
    char password[MAX_USERNAME_LENGTH];
    char currentCharacter;
    int i;
    for (i = 0; e.username[i] != '\0'; i++)
    {
        currentCharacter = toupper(e.username[i]);
        if(currentCharacter >= 'a' && currentCharacter <= 'z')
        {
            currentCharacter = currentCharacter + e.facultyKey;
            if(currentCharacter > 'z')
            {
                currentCharacter = currentCharacter - 'z' + 'a' - 1;
            }
            password[i] = currentCharacter;
        }
        else if (currentCharacter >= 'A' && currentCharacter <= 'Z')
        {
            currentCharacter = currentCharacter + e.facultyKey;
        }
        if(currentCharacter > 'Z')
        {
            currentCharacter = currentCharacter - 'Z' + 'A' - 1;
        }
        password[i] = currentCharacter;
    }
    printf("Encrypted Password is: %s\n", password);
}
```

```
void Caesar_Cipher_Decryption(char password[MAX_USERNAME_LENGTH], int facultyKey)
{
    char currentCharacter;
    int i;
    for (i = 0; password[i] != '\0'; i++)
    {
        currentCharacter = password[i];
        if(currentCharacter >= 'a' && currentCharacter <= 'z')
        {
            currentCharacter = currentCharacter - facultyKey;
            if(currentCharacter < 'a')
            {
                currentCharacter = currentCharacter + 'z' - 'a' + 1;
            }
            password[i] = currentCharacter;
        }
        else if (currentCharacter >= 'A' && currentCharacter <= 'Z')
        {
            currentCharacter = currentCharacter - facultyKey;
        }
        if(currentCharacter < 'A')
        {
            currentCharacter = currentCharacter + 'Z' - 'A' + 1;
        }
        password[i] = tolower(currentCharacter);
    }
    printf("Decrypted password is: %s\n", password);
}
```

On the other hand, Jose wrote the Decryption function which turned the encrypted text back to the original text with the one condition being that the key needs to be the same.

Problems and Solutions

Our group faced the challenge of contacting each other and get it touch. As two of our group members has not respond to either MS team or email, it makes it really hard to implement this group project with only two people. We overcome this challenge by making the project smaller and less complex so that it can be achieved by two people. Charlie and Jose frequently message each other in MS team in order to track the progress on the project and discuss the tasks to be completed.

Summary

A straightforward form of message encoding is the Caesar cipher. Caesar ciphers use a substitution approach in which the alphabetic characters are moved a predetermined number of times to produce an encoding alphabet. An A would be encoded as a B in a Caesar cipher with a 1 shift, a M as an N, a Z as an A, etc. The downside of Caesar cipher is that it has a simple structure which leads to minimum security. Moreover, the frequency of the letter can add clue to deciphering back to the original text. It is why Caesar Cipher is not used nowadays. More secure encryption program such as DES, AES, and RSA are being used widely nowadays on the internet which provides higher security. We looked at how Caesar cipher works in mathematics as well as how to implement it in C. Although we faced challenges such as unable to contact two group members, but we overcome this challenge by reducing the scope of our project and splitting the work evenly while keep in touch with other group member to track the progress. We have successfully implemented a simple Caesar Cipher Program in the language of C.

Appendices/Bibliography

- *C program to encrypt and decrypt files*. Available at: https://codescracker.com/c/program/c-program-encrypt-file.htm (Accessed: November 5, 2022).
- *Caesar - CS50X 2022* (no date) *Caesar - CS50x 2022*. Available at: https://cs50.harvard.edu/x/2022/psets/2/caesar/ (Accessed: November 5, 2022).
- *Caesar cipher* (2022) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Caesar_cipher (Accessed: November 5, 2022).
- dCode (no date) *Caesar cipher (shift) - online decoder, encoder, solver, translator*, *(Shift) - Online Decoder, Encoder, Solver, Translator*. Available at: https://www.dcode.fr/caesar-cipher (Accessed: November 5, 2022).
- Mishra, N. (2017) *Caesar cipher in C and C++ [encryption & decryption]*, *The Crazy Programmer*. Available at: https://www.thecrazyprogrammer.com/2016/11/caesar-cipher-c-c-encryption-decryption.html (Accessed: November 5, 2022).