

Last Submission Date: March 17, 2015 – 11:00am

T.A. Emre ASLAN, E-mail: aslan@bilmuh.gyte.edu.tr, Office : 123 Robotic Lab

	Q1	Q2	Q3
Points	30 points	20 points	50 points

GTU Encoding Table		<p>Assume that we are in a terrible science war. All universities sniff each other's network and try to understand what they are doing. In the cold science war you are expected to send messages not as plain text but as encrypted text.</p> <p>Yes, you are the chosen one and all you have and need is GTU encoding table and your amazing coding skills! Table is given left... Note that it only includes a part of the alphabet. The remaining characters are not going to be encrypted.</p> <p>Objectives</p> <ol style="list-style-type: none"> 1. Read the plain text message from PlainMessagesToSent.txt file 2. Write the encoded message to EncodedMessages.txt file 3. Crypt encoded messages by reading encoded messages file and changing each '1' with '*' and '0' with '_', also add '-' symbol according to following rule: Add a '-' symbol after each M characters. M starts with N and decreased by one after adding a '-' symbol until M is equal to zero. When M is equal to zero reinitialize it by N. N is given to be 5. 4. Write the encrypted message (encrypted by '*' '_' '-') to EncryptedMessages.txt file
Char	Code	
E	0	
I	10	
Space	110	
T	1110	
C	11110	
N	111110	
A	1111110	
G	11111110	
B	111111110	
Z	1111111110	
H	11111111110	
L	111111111110	
U	1111111111110	
V	11111111111110	
R	111111111111110	
S	1111111111111110	
Y	11111111111111110	

[illegible]

```
graph LR; subgraph ENCODING; direction TB; E1[OPEN FILES IN MAIN  
SEND FILE* TO  
encode_message(...) function]; E2[CALL  
encode_and_write_  
to_file(...) IN  
encode_message  
function to encode  
a character]; end; subgraph ENCRYPTION; direction TB; E3[OPEN FILES IN MAIN  
SEND FILE* TO  
crypt_message(...) function]; end;
```

ENCODING

OPEN FILES IN MAIN
SEND FILE* TO
encode_message(...) function

CALL
encode_and_write_
to_file(...) IN
encode_message
function to encode
a character

ENCRYPTION

OPEN FILES IN MAIN
SEND FILE* TO
crypt_message(...) function

GIVEN	WANTED
Template_1.c	Part1.c
PlainMessagesToSent.txt	EncodedMessages.txt
	CryptedMessages.txt

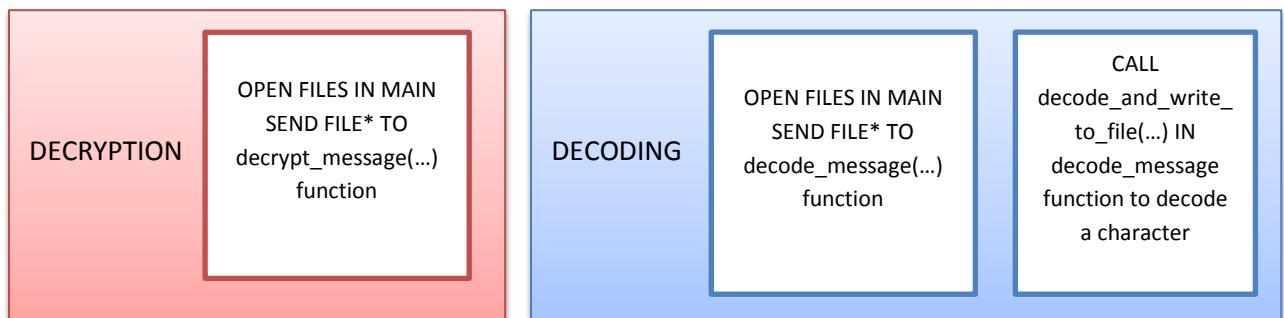
Note that you are expected to use Template_1.c file for Q1 which is attached to homework. You are going to find detailed info there also.

Q2) In this question, you are expected to read an encrypted message and convert it back to the plain text message.

Objectives

1. Read the encrypted message from **EncryptedInput.txt** which is attached to homework.
2. Eliminate '-' symbols and change '*' to '1' and '_' to '0' to obtain the encoded message. Write encoded message to **EncodedInput.txt** file.
3. Read the encoded input from **EncodedInput.txt** file.
4. Decode the encoded message and write the plain text message to **ReceivedMessage.txt** file.

Big Picture



GIVEN	WANTED
Template_2.c	Part2.c
CryptedInput.txt	EncodedInput.txt
	ReceivedMessage.txt

Note that you are expected to use Template_2.c file for Q2 which is attached to homework. You are going to find detailed info there also.

Q3) Now it is time to decode X University's messaging system! Assume X University uses a heuristic to create encoding table. Their heuristic is just based on frequency of letters. Frequently used letters have shorter code length.

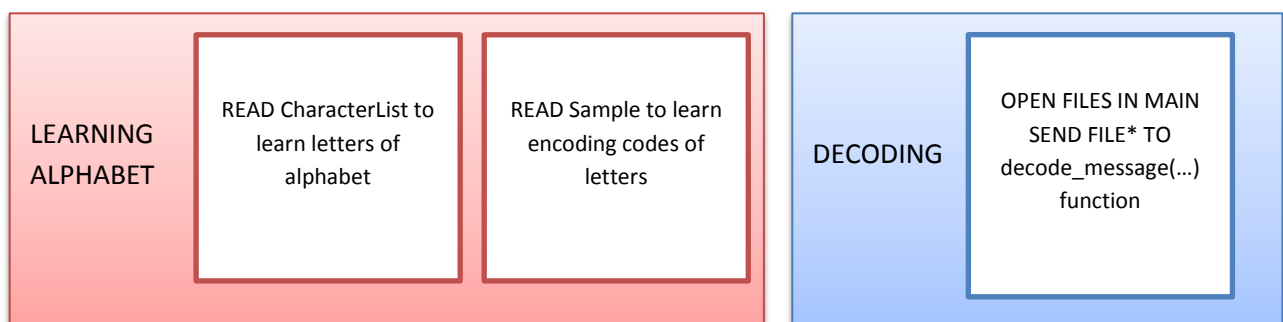
Example

Character List	A B C								
Sample	AABABAABCBAAC								
Frequency List of Sample	<table><tr><td>Char</td><td>A</td><td>B</td><td>C</td></tr><tr><td>Counts</td><td>7</td><td>4</td><td>2</td></tr></table>	Char	A	B	C	Counts	7	4	2
Char	A	B	C						
Counts	7	4	2						
Encoding Table of Sample	<table><tr><td>Char</td><td>A</td><td>B</td><td>C</td></tr><tr><td>Code</td><td>0</td><td>10</td><td>110</td></tr></table>	Char	A	B	C	Code	0	10	110
Char	A	B	C						
Code	0	10	110						
Encoded Message of X University	11010010011000110								
Plain Text Message of X University	CBABACAAC								

Objectives

1. You are given **CharacterList.txt** including three characters which are the letters of messaging system's alphabet. However, since the file is recovered from their trash it is broken (it contains some extra characters which are not a part of their alphabet). You should ignore all characters which are not letter like '!', '=', '+' etc. Write a function to figure out three letters and return number of letters read.
2. Read the file **Sample.txt** and count letters to find their frequencies and sort them to according to frequency. Implement swap functions given in template c file. You are going to use swap functions while sorting numbers and characters.
3. Since you figured out frequency of letters thanks to count letters function, you can handle decoding process. Decode the messages in **XUniversityEncoded.txt** file. Write the decoded (plain text) message to **XUniversityMessage.txt** file

Big Picture



GIVEN	WANTED
Template_3.c	Part3.c
CharacterList.txt	XUniversityMessage.txt
Sample.txt	
XUniversityEncoded.txt	

Note that you are expected to use Template_3.c file for Q3 which is attached to homework. You are going to find detailed info there also.

General:

1. Obey honor code principles.
2. Obey coding convention.
3. Read your homework carefully and follow the directives about the I/O format (data filenames, file formats, etc.) and submission format strictly. Violating any of these directives will be penalized.
4. Your submission should include the following file and NOTHING MORE (no data files, object files, etc):
HW04_<student name>_<student surname>_<student number>_part1.c
HW04_<student name>_<student surname>_<student number>_EncodedMessages.txt
HW04_<student name>_<student surname>_<student number>_CryptedMessages.txt
HW04_<student name>_<student surname>_<student number>_part2.c
HW04_<student name>_<student surname>_<student number>_EncodedInput.txt
HW04_<student name>_<student surname>_<student number>_ReceivedMessage.txt
HW04_<student name>_<student surname>_<student number>_part3.c
HW04_<student name>_<student surname>_<student number>_XUniversityMessage.txt
5. Do not use non-English characters in any part of your homework (in body, file name, etc.).
6. Deliver the printout of your work until the last submission date.