Autumn 2022, ISTM, Purdue-FCU 2+2 ECE Program
ISTM116 Programming Applications for Engineers, Final Exam

Use file name fexam_dxxxxxxx_1.c for Question 1, file name fexam_dxxxxxxx_2.c for Question 2, and file name fexam_dxxxxxxx_3.zip/rar/7z for Question 3 of your solutions, where dxxxxxxx is your student ID. When you finish a question, submit the above files to the instructor's computer.

1. (30 points) You may start with program skeleton **fexam_skeleton_1.c** and change the file name to **fexam_dxxxxxxx_1.c**. A substitutional Vigenère square is a variance of Vigenère square combining with substitution cipher. The first row is a substitution cipher code book and each of the following row is the cyclic left rotation of the row right on the top of it. An example of a substitutional Vigenère square is shown as the following:

Normal Letters	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 Letters	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M
	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N
	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E
	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I
	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q
	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O
	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y
	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A
	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R
	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D
	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C
	H	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S
Cyclic Left Rotation of the above Row	X	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H
	Z	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X
	B	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z
	P	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B
	J	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P
	T	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J
	K	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T
	U	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K
	F	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U
	L	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F
	V	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L
	G	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V
	W	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G
	M	N	E	I	Q	O	Y	A	R	D	C	S	H	X	Z	B	P	J	T	K	U	F	L	V	G	W

A keyword can be any English word which is repeated until the same length as the encoded text. The encoded text and the repeated keyword is aligned and then a code to book is selected. Selection of the code book is to match the aligned letter of the keyword with the fist letter of the code book. Note that code book selection is determined by the *inverse mapping* of the code book, i.e., the decode book. With the selected code book, the letter of the encoded text is then translated to a ciphered letter. For example, if the keyword is "FENGCHIA" and the text is "STRUCTUREPROGRAMMINGDESIGNINC" (Structure Programming Design in C), the encoding of the text is shown as below.

Input Text	S	T	R	U	C	T	U	R	E	P	R	O	G	R	A	M	M	I	N	G	D	E	S	I	G	N	I	N	C
Keyword	F	E	N	G	C	H	I	A	F	E	N	G	C	H	I	A	F	E	N	G	C	H	I	A	F	E	N	G	C
Encoded Text	X	F	T	T	H	O	V	G	W	J	Z	X	P	I	I	K	A	C	Z	Q	X	P	F	B	N	B	D	S	H

The encoded text is "XFTTHOVGWJZXPIIKACZQXPFBNBDSH". Write a C program to perform the following steps:

i. Input a keyword and an English text;
ii. Output the keyword and the original text;
iii. Create the decode book;
iv. Print the first code book and the first decode book.
v. Remove white spaces and punctuation symbols in the English text and convert all lower case letters to upper case letters;
vi. Encode the text using the Vigenère square with the code book given in the above substitutional Vigenère square and output the encoded text;
vii. Decode the encoded text and output the decoded text.

Program execution example:

```
Enter a keyword: FENGCHIA

Enter a line of English text: Structure Programming Design in C

**** The keyword is: FENGCHIA
>>>> The original text: Structure Programming Design in C

The first code book:
    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
    N E I Q O Y A R D C S H X Z B P J T K U F L V G W M

The first decode book:
    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
    G O J I B U X L C Q S V Z A E P D H K R T W Y M F N

>>>> The encoded text: XFTTHOVGWJTHPIIKACZQXPFBNBDSH

>>>> The decoded text: STRUCTUREPROGRAMMINGDESIGNINC
```

(to be continued)

2. (35 points) You may start with program skeleton **fexam_skeleton_2.c** and change the file name to **fexam_dxxxxxxx_2.c**. Let A be an n×n square matrix and X and C be column vectors of length n:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-2} & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-2} & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2,0} & a_{n-2,1} & \cdots & a_{n-2,n-2} & a_{n-2,n-1} \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-2} & a_{n-1,n-1} \end{bmatrix}, X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix}, \text{ and } C = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix}$$

Equation AX=C is exactly an n-variable linear system of equations as below:

$$\begin{aligned} a_{0,0}x_0 + a_{0,1}x_1 + \cdots + a_{0,n-2}x_{n-2} + a_{0,n-1}x_{n-1} &= c_0 \\ a_{1,0}x_0 + a_{1,1}x_1 + \cdots + a_{1,n-2}x_{n-2} + a_{1,n-1}x_{n-1} &= c_1 \\ &\vdots \\ a_{n-2,0}x_0 + a_{n-2,1}x_1 + \cdots + a_{n-2,n-2}x_{n-2} + a_{n-2,n-1}x_{n-1} &= c_{n-2} \\ a_{n-1,0}x_0 + a_{n-1,1}x_1 + \cdots + a_{n-1,n-2}x_{n-2} + a_{n-1,n-1}x_{n-1} &= c_{n-2} \end{aligned}$$

The solution of this linear system of equations using the determinant approach is,

$$x_0 = \frac{|replace(n, A, C, 0)|}{|A|}, \quad x_1 = \frac{|replace(n, A, C, 1)|}{|A|}, \cdots, \quad x_{n-1} = \frac{|replace(n, A, C, n-1)|}{|A|}.$$

if $|A| \neq 0$; otherwise, the linear system has no solution or many solutions, where $|A|$ is the determinant of matrix A and $|replace(n, A, C, i)|$ is the determinant of an n×n matrix after replacing the i-th column of matrix A by vector C. Write a C program using dynamic memory allocation to solve a system of linear equations. Perform the following steps:

i. Enter degree n of the linear system of equations.
ii. Allocate memory space for coefficient matrix A, constant vector C, and solution vector X.
iii. Use random number generator with seed 1000 to generate elements of A and C. Elements of A and C are between -1.0000 and 1.0000.
iv. Output matrix A, vector C, and the system of equations. Print floating point elements two digits after the decimal points.
v. Solve the linear system of equations using the determinant approach and output solution vector X.
vi. Verify the solution by checking the absolute value of all elements of vector AX-C being less than *epsilon*, where *epsilon* is 0.0001.

Program execution example:

```
Enter the degree of linear system (1 to 12): 6

>>>> Matrix A:
    -0.67 -0.18 -0.32  0.40 -0.85 -0.36
    -0.86 -0.30  0.02 -0.82 -0.34 -0.84
    -0.12  0.97 -0.73 -0.69  0.15  0.19
     0.36 -0.13 -0.19 -0.31  0.73 -0.45
    -0.16 -0.71 -0.85 -0.80 -0.45  0.48
    -0.25  0.50 -0.04 -0.60  0.67  0.94

>>>> Vactor C:
    -0.21  0.15 -0.96 -0.16  0.95  0.13


>>>> The linear system is:
    -0.67 X_0 -  0.18 X_1 -  0.32 X_2 +  0.40 X_3 -  0.85 X_4 -  0.36 X_5 = -0.21
    -0.86 X_0 -  0.30 X_1 +  0.02 X_2 -  0.82 X_3 -  0.34 X_4 -  0.84 X_5 = 0.15
    -0.12 X_0 +  0.97 X_1 -  0.73 X_2 -  0.69 X_3 +  0.15 X_4 +  0.19 X_5 = -0.96
     0.36 X_0 -  0.13 X_1 -  0.19 X_2 -  0.31 X_3 +  0.73 X_4 -  0.45 X_5 = -0.16
    -0.16 X_0 -  0.71 X_1 -  0.85 X_2 -  0.80 X_3 -  0.45 X_4 +  0.48 X_5 = 0.95
    -0.25 X_0 +  0.50 X_1 -  0.04 X_2 -  0.60 X_3 +  0.67 X_4 +  0.94 X_5 = 0.13

>>>> The solution of the linear system is:  0.02 -1.00  0.41 -0.35 -0.11  0.55

>>>> The solution is correct.
```

(to be continued)

3. (35 points) You may start with the project skeleton in directory **fexam_skeleton_3** and change the directory name to **fexam_dxxxxxxx_3**. Consider the expansion of $(a+b)^n$. If the coefficients of $(a+b)^k$ for $0 \le k \le n$, are listed in a row, we obtain a triangle, called **Yang-Hui triangle (楊輝三角形)**, also know as, **Pascal triangle**, as below for n=5:

k=0	1
k=1	1 1
k=2	1 2 1
k=3	1 3 3 1
k=4	1 4 6 4 1
k=5	1 5 10 10 5 1

Yang-Hui triangle can be solved using a queue. The following pseudo code is to print Yang-Hui triangle for a non-negative integer n with a queue Q:

1. Initialize queue Q.
2. Set k to 0.
3. If k<=n,
   - 3.1. Set $last$ = 0, i = 0;
   - 3.2 If i<k,
     - 3.2.1. $curr$ = $dequeue(Q)$;
     - 3.2.2. Print $last+curr$;
     - 3.2.3. $enqueue(Q, last+curr)$;
     - 3.2.4. $last$ = $curr$;
     - 3.2.5. i=i+1;
     - 3.2.6. Go to Step 3.2;
   - 3.3. Otherwise,
     - 3.3.1. $last$ = 1;
     - 3.3.2. Print $last$ and a newline;
     - 3.3.3. $enqueue(Q, last)$;
     - 3.3.4. k = k +1;
     - 3.3.5. Go to step 3.
4. Otherwise, clear queue Q.

Write a C project to implement Yang-Hui triangle using queue. Create a new C project named **fexam_dxxxxxxx_3.dev** in directory **fexam_dxxxxxxx_3** and add the three files **queue_fixed_array.h**, **queue_fixed_array.c**, and **YangHui_triangle_queue.c** in the project director. Complete the program of the project. After finishing the project, compress directory **fexam_dxxxxxxx_3** to get **fexam_dxxxxxxx_3.zip/rar/7z** and upload the compressed file.

Program execution examples:

```
>>>> Enter an integer (0 to 19, stop when less 0): 5
   1
   1   1
   1   2   1
   1   3   3   1
   1   4   6   4   1
   1   5  10  10   5   1
```

```
>>>> Enter an integer (0 to 19, stop when less 0): 10
   1
   1   1
   1   2   1
   1   3   3   1
   1   4   6   4   1
   1   5  10  10   5   1
   1   6  15  20  15   6   1
   1   7  21  35  35  21   7   1
   1   8  28  56  70  56  28   8   1
   1   9  36  84 126 126  84  36   9   1
   1  10  45 120 210 252 210 120  45  10   1
```