Course: EEE2020 Data Structure

Homework 1 **Due date: 2014.03.25. Before class starts**

Things you must do for this homework.

(1) **Upload *Report File* on YSCEC** (Report file should contain 'Flowchart', 'source code' and 'results')

(2) **Upload all *source codes* on YSCEC**.

(3) ***Print your Report File***, and ***hand it in*** before class

Late submissions will NOT be accepted.

This homework is not group assignment. Group submissions will NOT be accepted.

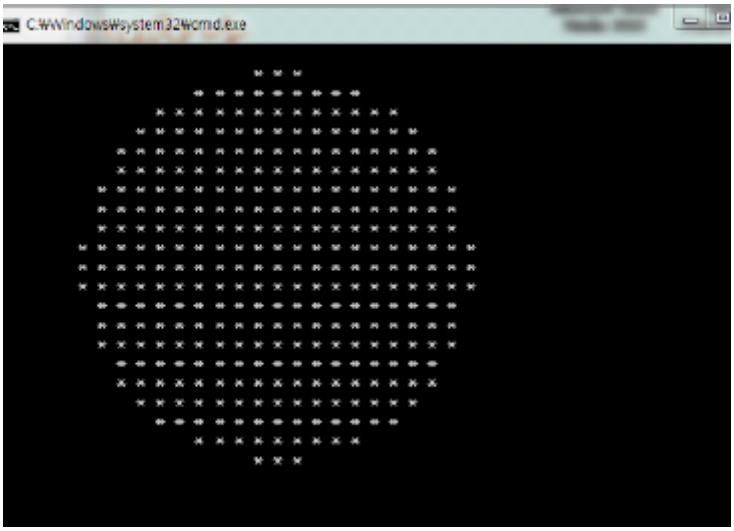
1. Write down a program to solve a simple arithmetic equation which consists of one operator, two natural numbers.

Input	Output
+ 1 2	3
- 4 5	-1
* 2 10	20
/ 5 10	0.2

2. Write down a program which converts upper-case letters to lower-case letters and lower-case letters to upper-case letters.

Input	Output
Biological Cybernetics	bIOLOGICAL cYBERNETICS

3. Write down a program which prints a circle. the user should type in the radius as input. For instance, if r = 10.5, the output should look like the figure below



4. Write down a program which returns one or two solutions of a second order equation. $ax^2 + bx + c = 0$. Variables (a, b and c) are user inputs.

Input	Output
1, 2, 1 0, 1, 2 1, 1, 1	x1 = -1, x2 = -1 x1 = -2 x1 =-0.5 +0.866i , x2 = -0.5 – 0.866i

5. Write down a program that read two integer numbers *m* , *n* and calculates the greatest common divisor (G.C.D.) and the least common multiple (L.C.M.)

Input	Output
6 8	G.C.D. : 2 L.C.M. : 24

6. Using the ‘for’ statement, write down a program which prints out the following result.

(a)

```
aaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbb
ccccccccccccccccccccc
ddddddddddddddddddddd
eeeeeeeeeeeeeeeeeeee
fffffffffffffffffff
ggggggggggggggggg
hhhhhhhhhhhhhhhh
iiiiiiiiiiiiiii
jjjjjjjjjjjjjj
kkkkkkkkkkkkkk
llllllllllllll
mmmmmmmmmmmmm
nnnnnnnnnnn
ooooooooooo
ppppppppp
qqqqqqqqq
rrrrrrrr
sssssss
tttttt
uuuuu
vvvv
www
yy
z
```

(b)

```
a
ab
abc
abcd
abcde
abcdef
abcdefg
abcdefgh
abcdefghi
abcdefghij
abcdefghijk
abcdefghijkl
abcdefghijklm
abcdefghijklmn
abcdefghijklmno
abcdefghijklmnop
abcdefghijklmnopq
abcdefghijklmnopqr
abcdefghijklmnopqrs
abcdefghijklmnopqrst
abcdefghijklmnopqrstu
abcdefghijklmnopqrstuv
abcdefghijklmnopqrstuvw
abcdefghijklmnopqrstuvwy
abcdefghijklmnopqrstuvwyz
```

7. The convolution operation is a mathematical operation of two discrete functions. This operation is defined by:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

The result, *y*[n], is typically viewed as a modified version of one of the original functions, giving the area overlap between the two functions as a function of the amount that one of the original functions is translated.
Write down a program to calculate the result of convolution operation for given two functions.
(1) Represent the following functions and store this in two arrays. Print out all values in each array using ‘printf’ statement. The size of each array is 100.

n = [0 … 99] , N = 100

- Rectangular functions:

$$x[n]\begin{cases} 1 & 25 < n < 75 \\ 0 & else \end{cases}, \quad h[n]\begin{cases} \frac{1}{2}n & 25 < n < 75 \\ 0 & else \end{cases}$$

(2) Write down a program to calculate the result of convolution operation for given functions in the above arrays, and save the result into an array. The length of the result may not be the same as the length of given functions.

Print out all values in the result array.

8. Fourier transform

We will test the Fourier transform at this homework.

An arbitrary temporal signal is saved in an array. The array includes 100 sample points. Then calculate the magnitude and phase of a specific frequency k as follows.

$$X[k] = a + bj = \sum_{n=o}^{N-1} x_n e^{-\frac{2\pi j}{N}kn}$$

where X is a complex number, x[n] is the n-th element of the input data signal, and N = 100, $e^{-\frac{2\pi j}{N}kn} = \cos(\frac{2\pi}{N}kn) - j\sin(\frac{2\pi}{N}kn)$, k = 0, 1, ..., N − 1. The magnitude is $\sqrt{a^2 + b^2}$ and the phase is atan2(b, a).

Test an input signal data $x_n = 3\sin\left(\frac{2\pi n}{N}\right) + 7\cos(\frac{3\pi n}{N})$ where n = 0, .., N − 1, and find the magnitude and phase of every frequency ranging from 0 to N − 1.

Test another input data is the 'rectangular' function we represent in the problem 7(1) (*x*[n]). You will check if the above program finds the dominant frequency over an arbitrary input signal.

You need to submit the input data, the result, the source code and a write-up document. The result should include the magnitude and phase for each frequency. A single program using for loop should test all the experiments together.

```
#include <stdio.h>
#include <math.h>
#define  N   100
#define  PI  3.1415926535
main()
{
float data[N];  // double declaration is possible
int m, n, k;
float real, imag; // double declaration is possible
float mag, phase;

//for(m=0; m < N; m++) data[m] = sin(2 *PI* 2*m / (float) N) +
2*cos(10*PI*m/ (float) N);
 for(m=0; m < N; m++) data[m] = sin(2 *PI* m / (float) N) ;

real = imag = 0.0;

scanf("%d",&k);
for(n=0; n < N; n++)  {
 real += cos(2*PI*k*n/ (float) N)  * data[n];
 imag += -sin(2*PI*k*n/ (float) N)  * data[n];
}
mag = sqrt(real*real + imag*imag);
phase = atan2(imag, real)/PI*180.0;

printf("%f %f %f %f\n", mag, phase, real, imag);
```