# Introduction to Computers and Programming LAB-Quiz3 2015/01/07

## Time: 2.5 hrs

**※Please create a new folder. Name the folder as: Student ID-Name (XXXXXXX-○○○). Inside the folder, your file format will be Q_1.c, Q_2.c, etc. There will ONLY be a total of 4 .c files in your folder (wrong file name or format will cause score deductions).**

**※No Internet. No discussions.**

**※The class is for C language, so do not use C++.**

**※If any of your program cannot be compiled, you will get zero score for the question.**

**※Your programs will be checked (by a tool) for the programming integrity. Be honest with your own works.**

1. (15 pts) Sum to One Digit

   Write a program that can sum each single digit number in the previous sequence until the result of sum becomes the single digit. For example, the initial sequence is 1234, then 1+2+3+4=>10, then keep sum the sequence 10, then 1+0=>1, so you should output 1.

```
Input the sequence: 1234
The result is 1

Input the sequence: 32654
The result is 2
```

2. (15 pts) Linked Lists

   Use dynamic storage allocation (malloc) to implement linked lists taught at Ch17 to store N elements of integer number which user input and print them all in order or reverse order.

   Note: using static storage in linklist is forbidden, TA will check your code.

```
Input N:5
store into the linklist:
1 2 3 4 5
print from the linklist:
5 4 3 2 1
```

3. (20 pts) Power Set

Please write a program that print power set of input array. The program will read array size and array and then print all the permutation combinations of this sequence. You can use recursive method, binary bit method or any other method.

Note: any order of output is OK

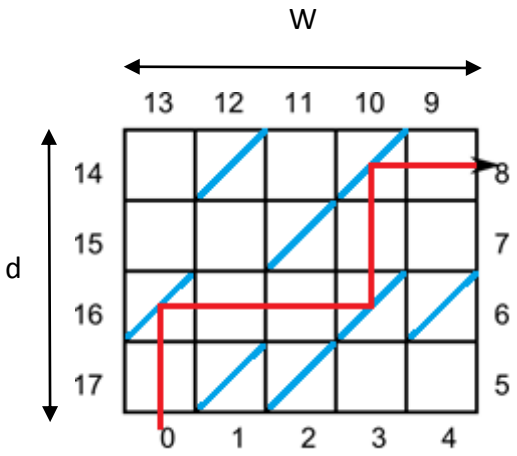
4. (20 pts) House Mirror

We divide a room into *d* by *w* spaces (i.e. denoted by a *d*×*w* array), where you can place a mirror or not. For example, in file 1.in, a room is divided into 4×5 spaces, where 1 indices a mirror, and 0 indicates no mirror.

File 1.in:

```
4 5
0 1 0 1 0
0 0 1 0 0
1 0 0 1 1
0 1 1 0 0
```

All the mirrors from 1.in are arranged as in the following figure.




We install the windows along the perimeter of the room, for example, there are 18 windows in 1.in, denoted by 0 to 17. In general, we label each window from 0 to 2(*w*+*d*) -1. Based on the arrangement of the mirrors in file 1.in, when we look into **window number 0**, we will see the view of **window number 8**, illustrated by the red line in figure above. Now given all the mirror positions, please compute the view for **each window** (0…2(*w*+*d*) -1), so when input is 1.in, you should output the view from 0 to 17 window.

Note: 1<=w<=100, 1<=d<=100

```
Input filename: 1.in
8 7 5 9 6 2 4 1 0 3 17 15 14 16 12 11 13 10
Input filename: 2.in
17 21 15 13 20 9 11 12 10 5 8 6 7 3 55 2 54 0 50 48 4 1 53 51 47 45 49 40 52 42
41 46 39 44 43 37 38 35 36 32 27 30 29 34 33 25 31 24 19 26 18 23 28 22 16 14
Input filename: 3.in
38 29 34 37 35 23 31 30 21 27 22 28 26 24 18 25 19 20 14 16 17 8 10 5 13 15 12 9
 11 1 7 6 48 47 2 4 46 3 0 49 45 44 43 42 41 40 36 33 32 39
```