



COMPUTER ENGINEERING

CHIANG MAI UNIVERSITY
CPE102 Computer Programming
Problem Set 15

1. จงเขียนฟังก์ชัน `sumNonZero()` ซึ่งเป็นฟังก์ชันสำหรับหาผลรวมของชุดตัวเลขที่ติดกัน โดยที่ตัวเลขทั้งหมดในช่วงนั้นไม่ได้มีค่าเท่ากับ 0 โดยฟังก์ชัน `sumNonZero()` มี Function Prototype ดังนี้

```
int sumNonZero(int *d,int **s,int **e)
```

- `d` เป็น Parameter ที่ใช้เป็นอินพุตระบุตำแหน่งเริ่มต้นของการค้นหาช่วงของชุดตัวเลขที่ไม่ใช่ 0 ที่ติดกัน
- `s` เป็น Parameter ที่ใช้ในการเก็บผลลัพธ์ซึ่งเป็นตำแหน่งเริ่มต้นของชุดตัวเลขที่พบ (ตัวเลขแรกที่ไม่ใช่ 0 ที่อยู่หลังจาก `d`)
- `e` เป็น Parameter ที่ใช้ในการเก็บผลลัพธ์ซึ่งเป็นตำแหน่งสุดท้ายของชุดตัวเลขที่พบ (ตัวเลขที่ไม่ใช่ 0 ตัวสุดท้ายที่ติดกันหลังจาก `s`)
- ฟังก์ชัน `sumNonZero()` จะ return ผลรวมของตัวเลขที่ไม่ใช่ 0 ที่ติดกันทั้งหมด ช่วงแรกที่พบ (ผลบวกตัวเลขตั้งแต่ตำแหน่ง `s` ถึง `e`)

Source Code (ให้เติมส่วนของ Function `sumNonZero` ในหน้าถัดไปให้ทำงานได้สอดคล้องกับผลลัพธ์และการเรียกใช้ใน `main`)

```
#include<iostream>
using namespace std;

int sumNonZero(int *,int **,int **);

int main(){

    int A[] = {0,0,0,0,0,1,3,2,1,1,0,0,0,1,6,9,0,0,0,1,1,0,0};
    int sum;
    int *start, *end;

    sum = sumNonZero(A,&start,&end);

    cout << "sum = " << sum << "\n";
    cout << "start = " << start << "\n" ;
    cout << "end = " << end << "\n";
    cout << "length = " << end-start+1 << "\n";
    cout << "-----\n";

    sum = sumNonZero(end+1,&start,&end);

    cout << "sum = " << sum << "\n";
    cout << "start = " << start << "\n" ;
    cout << "end = " << end << "\n";
    cout << "length = " << end-start+1 << "\n";
    cout << "-----\n";

    sum = sumNonZero(end+1,&start,&end);

    cout << "sum = " << sum << "\n";
    cout << "start = " << start << "\n" ;
    cout << "end = " << end << "\n";
    cout << "length = " << end-start+1 << "\n";
    cout << "-----\n";

}
```

```
int sumNonZero(int *d,int **s,int **e){  
    //Write definition of sumNonZero() here  
}
```

ตัวอย่างผลลัพธ์ที่ต้องการ

```
sum = 8  
start = 0x22fdd4  
end = 0x22fde4  
length = 5  
-----  
sum = 16  
start = 0x22fdf4  
end = 0x22fdfc  
length = 3  
-----  
sum = 2  
start = 0x22fe0c  
end = 0x22fe10  
length = 2  
-----
```

2. ให้นักศึกษาลองไปศึกษาวิธีการเรียงลำดับแบบ Quicksort ซึ่งสามารถสรุปเป็น Pseudo Code ได้ดังนี้

Pseudo Code ของ Quicksort Algorithm (<https://en.wikipedia.org/wiki/Quicksort>)

```
algorithm quicksort(A, lo, hi) is
    if lo < hi then
        p := partition(A, lo, hi)
        quicksort(A, lo, p - 1)
        quicksort(A, p + 1, hi)

algorithm partition(A, lo, hi) is
    pivot := A[hi]
    i := lo // place for swapping
    for j := lo to hi - 1 do
        if A[j] ≤ pivot then
            swap A[i] with A[j]
            i := i + 1
    swap A[i] with A[hi]
    return i

//Sorting the entire array is accomplished by quicksort(A, 1, length(A)).
```

โดยเราสามารถเขียน Source Code ภาษา C++ ที่สอดคล้องกับ Source Code ด้านบนได้ดังนี้ จงเติมส่วนที่ขาดหายไปให้สมบูรณ์ แล้วลองทดสอบนำไป Run ดู

Source Code

```
#include<iostream>
using namespace std;

template <typename T>
void quicksort(T *,T *);

template <typename T>
T * partition(T *,T *);

template <typename T>
void swap(T *,T *);

int main(){
    int data[10] = {7,2,4,5,6,7,0,1,9,3};
    int *p1 = &data[0];
    int *p2 = &data[9];

    quicksort(_____(A)_____);

    for(int *i = p1; i <= p2; i++){
        cout << *i << " ";
    }

    return 0;
}
```

```

template <typename T>
void quicksort(T *lo,T *hi){
    _____(B)_____ = partition(lo,hi);
    if(p-1 > lo) quicksort(lo,p-1);
    if(hi > p+1) quicksort(p+1,hi);
}

template <typename T>
T * partition(T *lo,T *hi){
    T pivot = _____(C)_____ ;
    _____(D)_____ = lo;
    for(_____(E)_____; j < hi;j++){
        if(_____(F)_____ <= pivot){
            swap(i,j);
            _____(G)_____ ;
        }
    }
    swap(i,hi);
    return i;
}

template <typename T>
void swap(T *x,T *y){
    T temp = *x;
    *x = *y;
    *y = temp;
}

```

ผลลัพธ์ที่ต้องการ

0	1	2	3	4	5	6	7	7	9
---	---	---	---	---	---	---	---	---	---

3. [Challenge Problem] จงเขียนโปรแกรมเพื่อสร้าง Array 2 มิติ โดยรับอินพุตจำนวนแถว N และ จำนวนหลัก M จากผู้ใช้ โดยเลขใน Array 2 มิติ นั้น เป็นเลขลำดับในการอ่านข้อมูล 2 มิติ แบบ Zig-zag Scan ซึ่งมีรูปแบบลำดับในการอ่านข้อมูลดังแสดงในรูปด้านล่าง

0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

ตัวอย่างการรับอินพุตและผลลัพธ์ที่ต้องการ (ตัวเลขที่ขีดเส้นใต้ คือ อินพุตจากผู้ใช้)

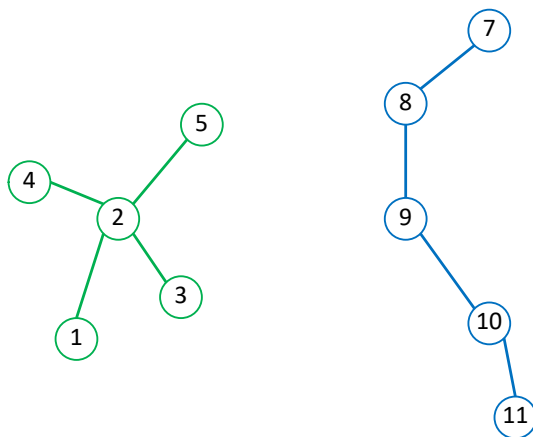
N, M = <u>6</u> <u>6</u>							
Reading sequence is:							
1	2	6	7	15	16		
3	5	8	14	17	26		
4	9	13	18	25	27		
10	12	19	24	28	33		
11	20	23	29	32	34		
21	22	30	31	35	36		

N, M = <u>4</u> <u>8</u>							
Reading sequence is:							
1	2	6	7	14	15	22	23
3	5	8	13	16	21	24	29
4	9	12	17	20	25	28	30
10	11	18	19	26	27	31	32

N, M = <u>12</u> <u>5</u>				
Reading sequence is:				
1	2	6	7	15
3	5	8	14	16
4	9	13	17	25
10	12	18	24	26
11	19	23	27	35
20	22	28	34	36
21	29	33	37	45
30	32	38	44	46
31	39	43	47	54
40	42	48	53	55
41	49	52	56	59
50	51	57	58	60

4. [Challenge Problem] จงเขียนโปรแกรมเพื่อหาจำนวนของกลุ่มดาว โดยสมมติให้ดาวแต่ละดวงมีเลขประจำตัวของมันซึ่งเป็นเลขจำนวนเต็มบวก ซึ่งอาจจะเป็นตัวเลขที่ซ้ำๆ หายๆ ไม่ได้เรียงต่อกัน โดยผู้ใช้จะอินพุตการเชื่อมต่อโดยใส่หมายเลขของดาวคู่ใดๆ ที่มีสายสัมพันธ์เชื่อมต่อกัน โดยดาวที่เป็นกลุ่มดาวเดียวกันนั้นหมายถึงดาวที่มีสายสัมพันธ์เชื่อมโยงไปถึงกัน เช่น จากตัวอย่างด้านล่างนี้ ประกอบด้วยดวงดาว 10 ดวงที่มีหมายเลข 1,2,3,4,5,7,8,9,10,11 (จะเห็นว่าตัวเลขไม่ต่อเนื่อง คือ ไม่มีหมายเลข 6) ผังขวามือด้านขวามือเป็นอินพุตที่ใช้กรอกสายสัมพันธ์ที่มีระหว่างดาวคู่ใด ๆ เมื่อผู้ใช้อินพุตสายสัมพันธ์ครบแล้ว สามารถออกจากโปรแกรมได้โดยการกรอกเลขใดเลขหนึ่งที่มีมากกว่าหรือเท่ากับ 0 โดยอินพุตในตัวอย่างนี้สามารถเขียนเป็นรูปได้ดังแสดงในรูปซ้ายมือ ซึ่งจะเห็นว่าอินพุตนี้ประกอบด้วยดวงดาว 2 กลุ่มแยกกันอยู่ ระบบก็จะตอบ

#Constellation = 2 ออกมา



```
Input link [1]: 1 2
Input link [2]: 2 3
Input link [3]: 4 2
Input link [4]: 5 2
Input link [5]: 8 9
Input link [6]: 7 8
Input link [7]: 10 9
Input link [8]: 10 11
Input link [9]: 0 0
```

ตัวอย่างการรับอินพุตและผลลัพธ์ที่ต้องการ (ตัวเลขที่ขีดเส้นใต้ คือ อินพุตจากผู้ใช้)

```
Input link [1]: 1 2
Input link [2]: 2 3
Input link [3]: 4 2
Input link [4]: 5 2
Input link [5]: 8 9
Input link [6]: 7 8
Input link [7]: 10 9
Input link [8]: 10 11
Input link [9]: 0 0
#Constellation = 2
```

```
Input link [1]: 1 2
Input link [2]: 5 6
Input link [3]: 10 20
Input link [4]: 15 10
Input link [5]: 1 15
Input link [6]: 7 6
Input link [7]: 18 19
Input link [8]: 19 25
Input link [9]: 25 26
Input link [10]: 26 27
Input link [11]: 0 0
#Constellation = 3
```

```
Input link [1]: 8 8
Input link [2]: 7 8
Input link [3]: 5 7
Input link [4]: 6 1
Input link [5]: 1 6
Input link [6]: 9 10
Input link [7]: 4 11
Input link [8]: -1 -69
#Constellation = 4
```