FACULTY OF ENGINEERING

CHULALONGKORN UNIVERSITY

2110211 Introduction to Data Structures

YEAR II, Second Semester, Midterm Examination, January 31, 2012, Time 13:00 – 16:00

ชื่อ-นามสกุล_____เลขประจำตัว | | | | | | | | | 2 | 1 | CR58_____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 11 ข้อในกระดาษคำถามคำตอบจำนวน 7 แผ่น 7 หน้า คะแนนเต็ม 95 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน สามารถใช้ดินสอเขียนคำตอบได้
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะวิศวกรรมศาสตร์
   มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (.....)

หมายเหตุ (เพิ่มเติม)

1. สำหรับข้อที่ให้ออกแบบ หรือ เขียนโปรแกรม คะแนนที่ได้จะแปรตามประสิทธิภาพในการทำงานของโปรแกรม
2. สำหรับข้อที่ให้วิเคราะห์เวลาการทำงาน คะแนนที่ได้จะแปรตามความใกล้เคียงความเป็นจริงของการวิเคราะห์
3. **นิสิตสามารถอ้างถึงและเรียกใช้คลาสต่าง ๆ ที่อยู่ในเอกสารประกอบการสอนได้โดยไม่จำเป็นต้องเขียนขึ้นมาใหม่**
4. **ในข้อที่ต้องออกแบบโครงสร้างข้อมูล นิสิตไม่จำเป็นต้องตรวจสอบถึงกรณีที่มีการใส่ข้อมูลเข้าไปมากกว่าเนื้อที่ที่มีอยู่ (เสมือนว่าการจองพื้นที่นั้นจองมากพอเสมอ) หรือ กรณีที่เอาข้อมูลออกเมื่อไม่มีข้อมูลอยู่ในโครงสร้างข้อมูล**
5. **ให้เขียนคำตอบลงในเฉพาะพื้นที่ที่เว้นว่างไว้**
6. ให้นิสิตเขียนรหัสประจำตัวและเลขที่ใน CR58 ในทุกหน้าของกระดาษคำถามด้วย

---

1. (5 คะแนน) กำหนดให้ a เป็นอาเรย์ขนาด 10 ช่องของข้อมูลประเภท int จงแสดงค่าของตัวแปร a[0], a[1], a[2], ... , a[9] ที่เป็นผลจากการทำงานของส่วนของโปรแกรมต่อไปนี้

```
int i,j,c,r;
int *a;
a = (int*) malloc(sizeof(int) * 10);
a[0] = 0; a[1] = 1; a[2] = 2; r = 3;
for (i = 2;i < 10;i++) {
  c = a[i];
  for (j = 0;j < c;j++)
    a[r++] = i;
}
```

.....

.....

.....

.....

.....

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

2. (5 คะแนน) สำหรับโครงสร้างข้อมูลประเภท Stack ถ้าเรากำหนดให้มีฟังก์ชันเพิ่มเติมชื่อ multiPop ดังแสดงในส่วนของโปรแกรมด้านล่างนี้ จงแสดงผลลัพธ์ของการทำงานของส่วนของโปรแกรมในฟังก์ชัน main ด้านล่างนี้ (สมมติให้ DType เป็นข้อมูลประเภท int)

```
void multiPop(Stack s, int k) {
  s->size = s->size - k;
  if (s->size < 0) s->size = 0;
}

int main(int argc, char *argv) {
  int i;
  Stack s = newStack(1);
  for (i = 0;i < 7;i++)
    push(s,i*10);
  top(s);
  multiPop(s,3);
  push(s,5);
  multiPop(s,2);
  push(s,1);
  printStack(s);
}
```

.....

.....

.....

.....

.....

3. (5 คะแนน) จงวิเคราะห์ประสิทธิภาพในการทำงานของส่วนของโปรแกรมต่อไปนี้ โดยให้ตอบเป็น Θ ของฟังก์ชันที่ขึ้นอยู่กับค่า n

```
int i,j;
int sum = 2;
for (i = 2;i < n;i = i*2){
  for (j = 0;j < n / 2;j++)
    sum++;
  for (j = n;j > n / 2;j--)
    sum++;
}
```

.....

.....

.....

.....

.....

4. (5 คะแนน) จงระบุว่าข้อย่อยต่อไปนี้ เป็นจริงหรือเป็นเท็จ

- $F(n) = n\sqrt{n} \in \Theta(n^{2.5})$ .....
- $F(n) = 14n + 1/n \in O(n^2)$ .....
- $F(n) = \sin(n) \in \Theta(n)$ .....
- $F(n) = (4n^{2.5} - n^{10}) \in O(n^{2.5})$ .....
- $F(n) = \log(n)\,(n + \log(n)) \in O(n \log n)$ .....

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

5. (5 คะแนน) จงเขียนฟังก์ชัน transpose(int **a, int r, int c) ซึ่งรับข้อมูลนำเข้าเป็นอาเรย์สองมิติที่มีขนาด r แถวและ c คอลัมน์ แล้วทำการสร้างอาเรย์สองมิติตัวใหม่ที่มีขนาด c แถว r คอลัมน์โดยที่อาเรย์ตัวใหม่นี้มีข้อมูลเป็น transpose ของข้อมูลตัวเก่า (สมมติให้ b เป็นอาเรย์ตัวใหม่ b[y][x] จะมีค่าเท่ากับ a[x][y])

```
int** transpose(int **a, int r, int c) {




}
```

6. (10 คะแนน) จงเขียนฟังก์ชัน scatterArrayList(ArrayList a,int *pos) ซึ่งจะทำการเรียงสับเปลี่ยนตำแหน่งของข้อมูลใน ArrayList a โดยตัวแปร pos จะเป็นตัวแปรซึ่งบอกตำแหน่งของการเรียงสับเปลี่ยน ตัวแปร pos เป็นอาเรย์ 1 มิติที่มีจำนวนช่องเท่ากับจำนวนข้อมูลของ a และ 0 <= pos[i] < จำนวนข้อมูลใน a ถ้าเดิม ตำแหน่งที่ i ของ a เก็บ x หลังเรียก scatterArrayList จะทำให้ x ย้าย ไปเก็บที่ตำแแหน่ง pos[i] ของ a และ**pos[i] แต่ละตัวแตกต่างกัน** ยกตัวอย่าง เช่น ให้ a มีข้อมูลเป็น [10 20 30 40] และ pos มีข้อมูลเป็น [1 2 0 3] ผลลัพธ์ของการเรียงสับเปลี่ยนจะทำให้ a มีข้อมูลเป็น [30 10 20 40]
ในข้อนี้ **ห้ามนิสิตทำการสร้างอาเรย์ขึ้นมาใหม่** นิสิตสามารถสร้างตัวแปรประเภท DType ขึ้นมาได้ไม่จำกัด แต่ห้ามสร้างอาเรย์ของ DType ขึ้นมาโดยเด็ดขาด และจงวิเคราะห์ประสิทธิภาพในการทำงานของฟังก์ชันที่เขียนขึ้น

```
void scatterArrayList(ArrayList a, int *pos) {




}
```

7. (10 คะแนน) จงเขียนฟังก์ชัน isReverseArrayList(ArrayList a, ArrayList b) สำหรับโครงสร้างข้อมูลประเภท ArrayList เพื่อทำการตรวจสอบว่า ArrayList 2 อันที่รับมานั้นเป็น reverse ของกันและกันหรือไม่ โดยนิยามให้ ArrayList 2 อันเป็น reverse ของกันและกันก็ต่อเมื่อ ArrayList ทั้งสองมีจำนวนข้อมูลเท่ากันและมีข้อมูลเหมือนกันแต่เรียงลำดับสลับกันจากหัวมาท้าย โดย isReverseArrayList จะคืนค่า 0 ก็ต่อเมื่อ ArrayList ทั้งสองไม่ได้เป็น reverse เท่านั้น
ตัวอย่างเช่น ArrayList ที่มีข้อมูลเป็น [1 2 3] เป็น reverse ของ [3 2 1] แต่ [1 2 3] ไม่ได้เป็น reverse ของ [3 2 1 0] และไม่ได้เป็น reverse ของ [3 1 2] เป็นต้น และจงวิเคราะห์ประสิทธิภาพในการทำงานของฟังก์ชันที่เขียนขึ้น **หมายเหตุ: การเรียกใช้งาน isReverseArrayList นั้น เมื่อทำงานเสร็จแล้วข้อมูลใน ArrayList ทั้งสองจะต้องเหมือนเดิมก่อนการเรียกใช้งาน**

```
int isReverseArrayList(ArrayList a, ArrayList b) {




}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

8. (10 คะแนน) สำหรับโครงสร้างข้อมูลแบบ Queue นั้น การนำข้อมูลออกมาจะต้องเป็นไปตามลำดับคิว คือเข้าก่อนออกก่อน จงเขียนฟังก์ชัน moveToFrontQueue(Queue q, int idx) เพื่อทำการ “ลัดคิว” โดยนำข้อมูลที่อยู่เป็นลำดับที่ idx จากหัวคิว (ให้หัวคิวมีลำดับเป็น 0) ย้ายมาแทรกอยู่เป็นตัวแรกของ Queue (คือย่ายมาเป็นหัวคิว) โดยข้อมูลอื่นยังอยู่ในคิวตามลำดับเดิม **(หมายเหตุ: รับประกันว่า idx นั้นจะมีค่าไม่น้อยกว่า 0 และน้อยกว่าจำนวนข้อมูลใน Queue เสมอ)**

```
void moveToFrontQueue(Queue q, int idx) {




}
```

9. (10 คะแนน) โครงสร้างข้อมูลประเภท Stack นั้น ใน struct SStack มีตัวแปรอยู่สามตัวคือ data, size และ length จงเขียนโครงสร้างข้อมูลประเภท Stack ขึ้นใหม่ โดยกำหนดให้ใน struct SStack นั้นแทนที่จะมีตัวแปรดังกล่าว กลับมีโครงสร้างข้อมูลประเภท Queue อยู่แทน (สมมติให้มีฟังก์ชันและส่วนของโปรแกรมต่าง ๆ ตามที่มีอยู่ใน Queue อยู่แล้ว) โดยให้เขียนเฉพาะฟังก์ชัน newStack, push และ pop นอกจากนี้ยังมีข้อกำหนดเพิ่มเติมคือให้นิสิตใช้คำสั่งของ queue ดังต่อไปนี้ได้เท่านั้น newQueue, isEmptyQueue, sizeOfQueue, enqueue, และ dequeue โดยห้ามเรียกใช้ตัวแปร data, size, front, length ที่อยู่ภายใน struct SQueue โดยเด็ดขาด (หมายเหตุ: ไม่จำเป็นต้องตรวจสอบกรณีที่มีการใส่ข้อมูลเมื่อกองซ้อนนี้เต็ม หรือการเอาข้อมูลออกเมื่อกองซ้อนนี้ไม่มีข้อมูล **<u>ประสิทธิภาพการทำงานของ Stack แบบใหม่อาจไม่ดีเท่าแบบเดิมที่นำเสนอในชั้นเรียน แต่ ขอให้พยายามทำให้มีประสิทธิภาพที่สุด</u>**)

```
// กำหนดให้มีโครงสร้างข้อมูลประเภท Queue อยู่เรียบร้อยแล้ว ไม่ได้เขียนไว้ ณ ที่นี้ แต่สามารถเรียกใช้ได้ตามปรกติ
typedef float DType;
struct SStack {
  Queue q;      // ห้ามประกาศตัวแปรใด ๆ เพิ่มเติม
};
typedef struct SStack *Stack;
Stack newStack(int length) {



}
void push(Stack s, DType x) {









}
DType pop(Stack s) {









}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

10. (15 คะแนน) นิสิตได้เรียนโครงสร้างข้อมูลประเภท กองซ้อน (Stack) และ แถวคอย (Queue) ซึ่งมีประโยชน์มาก แต่ทว่าในบางกรณีการที่เรามีโครงสร้างข้อมูลที่เป็นได้ทั้ง กองซ้อน และ แถวคอย ในตัวเดียวกัน ซึ่งเรียกว่าแถวคอยสองด้าน(Deque) ก็จะยิ่งมีประโยชน์มากขึ้น ในปัญหาข้อนี้จะให้นิสิตเขียนโปรแกรมของโครงสร้างข้อมูลนี้โดยเติมในช่องว่างด้านล่าง ภายในฟังก์ชั่นดังต่อไปนี้
    - void pushFront(Deque d, DType x) ใช้เพื่อใส่ x เข้าไปด้านหน้าของแถวคอยสองด้าน
    - void pushBack(Deque d, DType x) ใช้เพื่อใส่ x เข้าไปด้านหลังของแถวคอยสองด้าน
    - DType popFront(Deque d) ใช้เพื่อลบข้อมูลที่อยู่ด้านหน้าของแถวคอยสองด้านออกและคืนค่าข้อมูลนั้น
    - DType popBack(Deque d) ใช้เพื่อลบข้อมูลที่อยู่ด้านหลังของแถวคอยสองด้านออกและคืนค่าข้อมูลนั้น
    - void printDeque(Deque d) ใช้เพื่อแสดงข้อมูลที่อยู่ในแถวคอยสองด้านโดยแสดงตามลำดับจากด้านหน้าไปหลัง

    (ไม่จำเป็นต้องตรวจสอบกรณีที่มีการใส่ข้อมูลเมื่อแถวคอยสองด้านนี้เต็ม หรือการเอาข้อมูลออกเมื่อแถวคอยสองด้านนี้ไม่มีข้อมูล แต่ว่าโปรแกรมจะต้องมองอาเรย์เป็นวงวนเพื่อที่จะใช้เนื้อที่ได้เต็มที่ และคำสั่งpushFront, pushBack, popFront, popBack จะต้องใช้เวลาการทำงานเป็น O(1))

```
typedef float DType;
char *toString(char *buf, float d) {
    sprintf(buf, "%6.2f", d);
    return buf;
}
struct SDeque{
   DType* data;
   int length;  // ความจุของแถวคอยสองด้าน
   int size;     // จำนวนข้อมูลที่มีอยู่ตอนนี้
   int front; // ตำแหน่งของข้อมูลที่อยู่หน้าสุด
};
typedef SDeque* Deque;
Deque newDeque(Deque d, int length) {
   d = (struct SDeque*)malloc(sizeof(Deque));
   d->data = (DType*)malloc(sizeof(DType)*length);
   d->length = length;
   d->size = 0;
   d->front = 0;
}
void printDeque(Deque d)     {









}
void pushFront(Deque d, DType x) {









}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

```
void pushBack(Deque d, DType x) {




}
DType popFront(Deque d) {




}
DType popBack(Deque d) {




}
```

11. (15 คะแนน) ในโจทย์ข้อนี้ นิสิตจะต้องออกแบบโครงสร้างข้อมูลประเภทใหม่ โดยมีเป้าหมายเพื่อใช้งานในระบบร้านขายของออนไลน์ สมมติให้คุณเป็นเจ้าของร้านออนไลน์ร้านหนึ่ง ซึ่งขายของอยู่ K ประเภท สมมติให้ของแต่ละประเภทมีชื่อเรียกว่า 0, 1, 2, ..., k-1 เมื่อเปิดร้าน ของแต่ละชนิดจะมีอยู่ 0 ชิ้น เมื่อผู้ผลิตส่งของมาที่ร้านของคุณ คุณจะต้องทำการปรับจำนวนสินค้าที่อยู่ภายในร้าน โดยผู้ผลิตจะส่ง**ใบส่งของ** ซึ่งเป็นอาเรย์ของประเภทของสินค้าที่ส่งมาที่ร้านของคุณ (เช่น อาเรย์ที่มีข้อมูล [3,0,0,2] แปลว่าผู้ผลิตส่งของหมายเลข 3 หนึ่งชิ้น ของหมายเลข 0 สองชิ้นและของหมายเลข 2 หนึ่งชิ้น มาให้คุณ)
การขายของนั้นทำโดยสั่งผ่าน web โดยที่ผู้ใช้จะส่ง**ใบสั่งซื้อ** ซึ่งเป็นอาเรย์ของของที่ต้องการซื้อในรูปแบบเดียวกับใบส่งของ (เช่น อาเรย์ที่มีข้อมูล [1,1,2] แปลว่าลูกค้าต้องการของหมายเลข 1 สองชิ้น ของหมายเลข 2 หนึ่งชิ้น)
ร้านของคุณจะทยอยส่งของตามลำดับของใบสั่งซื้อที่เข้ามา โดยมีพนักงานทำหน้าที่จัดของและส่งของ แต่ร้านของคุณมีพนักงานส่งของอยู่จำกัด ดังนั้นมันอาจจะมีใบสั่งซื้อเข้ามาในขณะที่พนักงานของคุณกำลังส่งของอยู่ก็เป็นได้ ซึ่งใบสั่งซื้อดังกล่าวจะต้องรอจนกระทั่งพนักงานส่งของพร้อมที่จะทำการส่ง เมื่อพนักงานพร้อมที่จะทำการส่ง พนักงานจะทำการตรวจสอบสต็อกของสินค้า ถ้าสินค้าที่ลูกค้าต้องการมีเพียงพอทุกชนิด พนักงานก็จะทำการส่งของ แต่ถ้ามีสินค้าแม้เพียงอย่างเดียวไม่เพียงพอ พนักงานจะยกเลิกใบสั่งสินค้าดังกล่าวพร้อมกับแจ้งให้ลูกค้าทราบ และไปทำงานกับใบส่งสินค้าถัดไปแทน
หน้าที่ของคุณคือออกแบบโครงสร้างข้อมูลชื่อ Store ให้นิสิตเขียนโปรแกรมของโครงสร้างข้อมูลนี้ ซึ่งต้องมีฟังก์ชั่นดังต่อไปนี้อยู่

- Store newStore(int k) ทำหน้าที่สร้างโครงสร้างข้อมูล Store สำหรับร้านค้าที่มีสินค้า K ประเภท
- void supply(Store s, int *items, int n) เป็นฟังก์ชันในการรับสินค้าจากผู้ผลิต โดย**ใบส่งของ**ซึ่งมีข้อมูล n ตัวจะแสดงอยู่ในอาเรย์ชื่อ items
- void order(Store s, int *items, int n) เป็นฟังก์ชันสำหรับรับข้อมูลใบสั่งซื้อ โดย**ใบสั่งซื้อ**ซึ่งมีข้อมูล n ตัวจะแสดงอยู่ในอาเรย์ชื่อ items
- void process(Store s) เป็นฟังก์ชันที่พนักงานส่งของจะเรียกใช้เพื่อตรวจสอบว่าใบรายการสั่งซื้อที่ต้องทำการส่งมีรายละเอียดอย่างไร โดยฟังก์ชันนี้จะแสดงรายการสินค้าที่ต้องส่งออกมาทางหน้าจอ หรือแสดงคำพูดว่า “not enough stock” ถ้าสินค้าที่ต้องการมีไม่เพียงพอ

ให้นิสิตเขียนโปรแกรมลงในที่ว่างด้านล่างนี้ โดยนิสิตจะต้องเขียน struct สำหรับโครงสร้างข้อมูล Store และเขียนฟังก์ชันทั้ง 4 ตัวตามโจทย์ นิสิตสามารถเรียกใช้โครงสร้างข้อมูลที่เรียนมาแล้วได้ทุกอย่าง และสามารถปรับเปลี่ยน DType ของแต่ละโครงสร้างข้อมูลได้ โดยให้ระบุเพียงแค่ว่าใช้โครงสร้างข้อมูลอะไรบ้างในด้านล่างนี้ (ถ้าพื้นที่ไม่พอให้เขียนต่อด้านหลังได้)

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

รายการโครงสร้างข้อมูลที่ต้องการใช้ พร้อม DType ของโครงสร้างข้อมูลนั้น

.....

.....

```
struct SStore {
                                        //นิสิตสามารถเพิ่มเติมตัวแปรใด ๆ ภายในโครงสร้างข้อมูลนี้ได้



}

typedef struct SStore *Store;
```