ชื่อ _____ รหัส _____ section ___

มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 2 **ปีการศึกษา**: 2552

วันที่สอบ: 18 กุมภาพันธ์ 2553 **เวลาสอบ**: 9.00 – 12.00 น. **ห้องสอบ**: S101, S103

ผู้สอน: อ. อารีย์ ธีรภาพเสรี อ.เสกสรรค์ สุวรรณมณี อ.อัมรินทร์ ดีมะการ ภาควิชาวิศวกรรมคอมพิวเตอร์

รหัสและชื่อวิชา: **241-207** Data Structure and Computer Programming Techniques

โครงสร้างข้อมูลและเทคนิคการเขียนโปรแกรม

ทุจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และ**เอกสารใดๆ เข้าและออกห้องสอบ**

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 14 หน้า (*รวมใบปะหน้า*) มีทั้งหมด11 ข้อ คะแนนรวม 140 คะแนน (คิดเป็น 35%)
- **<u>เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด</u>**
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มที่ด้านหลังของหน้านั้นเท่านั้น

ข้อ	1 (30)	2 (10)	3 (13)	4 (7)	5 (5)	6 (22)	7 (8)	8 (10)	9 (15)	10 (16)	11 (4)	รวม (140)
คะแนน												

ชื่อ _____ รหัส _____ section ___

Linked List (30 คะแนน)

1. พิจารณาส่วนของโค้ดโปรแกรมภาษาซีนี้ และตอบคำถาม ข้อ 1.1 – 1.4

```
struct listnode {
         int data;
         struct listnode *nextptr;
};
typedef struct listnode  LISTNODE;
typedef LISTNODE *LISTP; // pointer to a list

// create a new list with value at the first node
LISTP createList(int value)
{    LISTP lst;
     lst = malloc(sizeof(LISTNODE));
     if (lst!=NULL) { lst->data=value;  lst->nextptr=NULL; }
     return lst;
}

// add a node with value at the end of the list
int add(LISTP *lst, int value)
{ LISTP newptr, prevptr, currentptr;
  if ( isEmpty(*lst) ) // make a new list with the value
  {  *lst=createList(value);
     if (*lst != NULL ) return 1;
     else return 0;
  }
  newptr=malloc(sizeof(LISTNODE));
  if (newptr==NULL) return 0; // no memery available
  newptr->data = value;
  newptr->nextptr = NULL;
  currentptr=(*lst);
  while(currentptr!=NULL)
  { prevptr=currentptr;
    currentptr=currentptr->nextptr;
  }
  prevptr->nextptr=newptr;
  newptr->nextptr=currentptr;
  return 1;
}
```

1.1) จงเขียนนิยามของฟังก์ชัน `isEmpty` ที่ตรวจสอบว่าลิสต์ที่รับมาในพารามิเตอร์ของฟังก์ชัน เป็นลิสต์ว่างหรือไม่ ถ้าเป็นลิสต์ว่างให้ คืนค่า 1 ถ้าไม่ใช่ คืนค่า 0 (5 คะแนน)

```
int isEmpty(LISTP list)
```

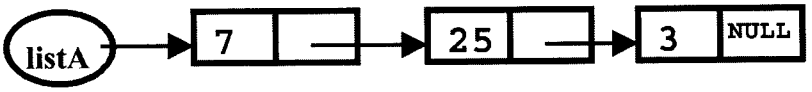
1.2) จงเขียนนิยามของฟังก์ชัน `cons` ที่แทรกโหนดใหม่ด้วยค่า `value` ตรงส่วนหัวของลิสต์ ให้ตรวจสอบก่อนว่าเป็นลิสต์ว่างหรือไม่ ถ้าเป็นลิสต์ว่างให้สร้างลิสต์ขึ้นมาใหม่ (เรียกใช้ฟังก์ชัน `createList`) ถ้าไม่ใช่ลิสต์ว่างจึงเพิ่มโหนดใหม่เข้าไป ฟังก์ชันคืนค่า 1 ถ้าทำได้สำเร็จ และคืนค่า 0 หากไม่สามารถแทรกโหนดใหม่ได้ (10 คะแนน)

ชื่อ _____ รหัส _____ section ___

```
//1.2) insert a value at the head of list
int cons(int value, LISTP *lst)
```

1.3) จงเขียนรูปแสดงลิสต์ listA ใหม่ หลังจากการทำงานของส่วนของโค้ดโปรแกรมนี้เสร็จสิ้น (10 คะแนน)

รูปแสดงลิสต์ listA




```
LISTP tmp;
if (!isEmpty(listA) ) {
      tmp = listA;
      listA = listA->nextptr;
      free(tmp);
}
cons(6, &listA);
add(&listA, 10);
```

1.4) จงเขียนนิยามของฟังก์ชัน numNode ที่คืนค่าเป็นจำนวน node ของลิสต์ จะใช้ recursion หรือ loop ก็ได้

(5 คะแนน)

```
int numNode(LISTP lst)
```

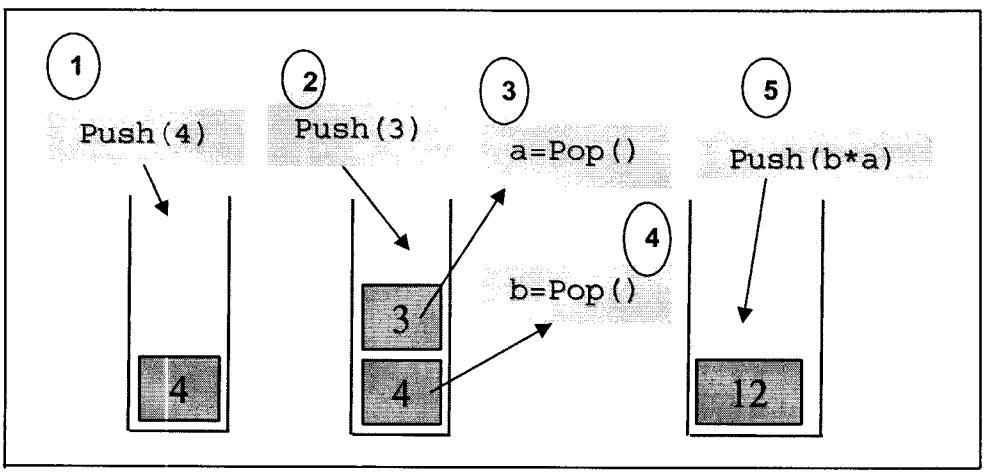
ชื่อ _____ รหัส _____ section ___

Stack and Queue (30 คะแนน)

2. จากคุณสมบัติหรือหลักการทำงานต่อไปนี้ ให้ทำเครื่องหมาย X ในช่องที่เหมาะสมที่สุด (10 คะแนน)

คุณสมบัติ หรือ หลักการทำงาน	Stack เท่านั้น	Queue เท่านั้น	Stack และ Queue	ไม่ใช่ทั้ง Stack และ Queue
สามารถ implement โดยใช้ linked list ได้				
สามารถ implement โดยใช้ array ได้				
LIFO (last in first out)				
FIFO (first in first out)				
เป็นโครงสร้างข้อมูลที่เหมาะกับการคำนวณค่านิพจน์ในรูปแบบ Reverse Polish Notation (Postfix)				
สามารถประยุกต์ใช้ Binary Search ได้กับโครงสร้างข้อมูลนี้ได้โดยง่าย				
เป็นโครงสร้างข้อมูลที่เหมาะกับการประยุกต์ใช้ในการเก็บลำดับการทำงาน ในลักษณะ Recursion				
เป็นโครงสร้างข้อมูลแบบ Non-linear				
มีการใช้การดำเนินการ(operation) Push กับ Pop				
ใช้ทำ Job scheduling แบบ First Come First Served				

3. จากรูปตัวอย่างข่างล่าง แสดงการใช้ Stack ในการคำนวณค่าของนิพจน์ `4 3 *` (เขียนแบบ Postfix Notation) ซึ่งจะได้ผลลัพธ์เป็น 12 (13 คะแนน)



ชื่อ _____ รหัส _____ section ___

จงเขียนรูป แสดงการคำนวณนิพจน์ พร้อมทั้งระบุลำดับของการดำเนินการ และผลลัพธ์สุดท้าย ของนิพจน์นี้

2 9 1 - 2 / + (แสดงให้อยู่ในรูปดังตัวอย่าง)

4. จงวาดรูปหรือไดอะแกรมเพื่ออธิบายโครงสร้างและส่วนประกอบที่สำคัญของคิว (Queue) ที่ implement โดยใช้ Singly Linked List (7 คะแนน)

ชื่อ _____ รหัส _____ section ___

Binary Tree (35 คะแนน)

5. จงเติมคำตอบ หรือ ตอบคำถามต่อไปนี้ (5 คะแนน)

5.1) Binary tree คือ โครงสร้างต้นไม้ที่ _____

5.2) Leaf node คือ _____

5.3) Binary tree ที่มีความสูง n สามารถมี leaf node มากที่สุดเป็นจำนวนเท่าไร ? _____

5.4) ใน Binary Search Tree ค่าที่มากที่สุด จะพบอยู่ที่ราก (tree root) เสมอ ใช่หรือไม่? _____
ถ้าไม่ใช่ ค่านั้นจะอยู่ที่โหนดใด? _____

5.5) การท่องไปในโครงสร้างต้นไม้ (Binary Tree Traversal) โดยเริ่มต้นจากรากแล้วท่องไปในทุกโหนดที่อยู่ในระดับเดียวกันก่อน จากซ้ายไปขวา จนไปถึงระดับสุดท้าย เรียกว่า _____

6. จากการท่องไปในโครงสร้าง binary tree BT_1 ซึ่งมี 8 โหนด โดยวิธี Preorder traversal และ Inorder traversal ได้พิมพ์ค่าแต่ละโหนดตามลำดับ ได้ดังนี้ (22 คะแนน)

```
Inorder traversal:  1 2 4 3 5 6 7 8
Preorder traversal: 5 2 1 3 4 7 6 8
```

จงตอบคำถามต่อไปนี้

6.1) BT_1 เป็น Binary Search Tree หรือไม่ ? _____

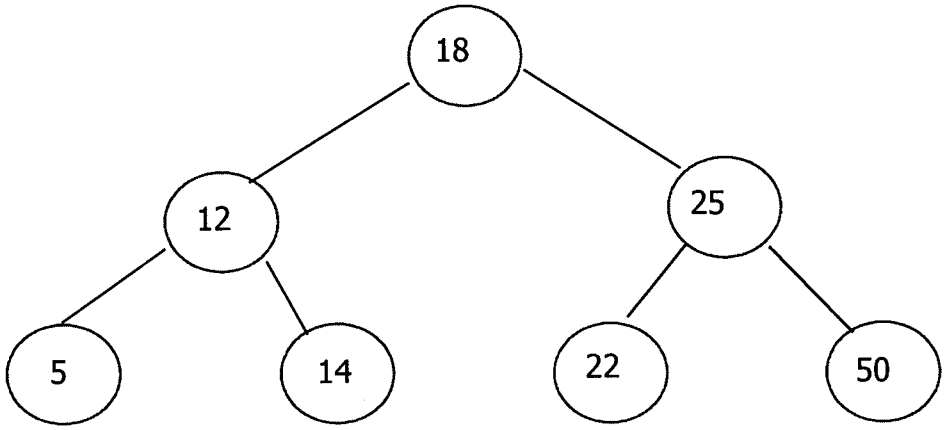
6.2) จงวาดรูปแสดงโครงสร้างของ BT_1

ชื่อ _____ รหัส _____ section ___

6.3) Leaf nodes ของ BT_1 คือโหนดใดบ้าง ? _____

6.4) BT_1 มีความสูง (Height) เท่าไร ? _____

7. Binary Search Tree BT_2 มีโครงสร้างดังรูป



จาก BT_2 เมื่อทำการลบโหนด 22 แล้ว แทรกโหนด 17 และ 20 ตามลำดับ จากนั้นจึงลบโหนด 14 จงวาดรูป Binary Search Tree ผลลัพธ์ (8 คะแนน)

ชื่อ _____ รหัส _____ section ___

Searching (25 คะแนน)

8. จงตอบคำถามต่อไปนี้ (10 คะแนน)

8.1) Big-O ของอัลกอริธึมในการค้นหาใช้เวลา $10n + n\log n + 20\log n$ คือ

_____

8.2) จากฟังก์ชันในการวัดประสิทธิภาพของโปรแกรมที่กำหนดให้ จงเรียงลำดับของค่า Big-O ต่อไปนี้ โดยเรียงลำดับจากประสิทธิภาพที่ดีที่สุดไปประสิทธิภาพที่ด้อยที่สุด

โดยเรียงลำดับจากฟังก์ชันต่อไปนี้ $O(2^n)$, O(n) , O(nlog n) , $O(n^2)$, O(1), O(log n)

O(X) –> O(Y) หมายถึง O(X) มีประสิทธิภาพดีกว่า โปรแกรม O(Y)

O(1) --> _____--> _____ --> _____ --> _____ --> _____

8.3) ค่าประสิทธิภาพเฉลี่ย (Average case performance) ของ linear search คือ

_____

8.4) Worst case ในการทำ Linear search เกิดขึ้นเมื่อใด

_____

8.5) ข้อมูลที่ใช้ในการค้นหาแบบ Binary Search จะต้องจัดเรียงข้อมูลก่อนหรือไม่ อย่างไร ?

_____

9. จงนำกลุ่มของคำตอบต่อไปนี้ ใช้ในการตอบคำถามข้อ 9.1 และ 9.2 (15 คะแนน)

กลุ่มของคำตอบ

```
A. array[index]==key
B. array[mid] == key
C. index+1
D. index>=size
E. mid=(low+high)/2
F. left < right
G. low=mid+1
H. mid
I. mid + 1
J. size - 1
```

ชื่อ _____ รหัส _____ section ___

9.1) จงเติมส่วนที่ขาดหายไป โดยเลือกจากกลุ่มของคำตอบที่กำหนดให้ เพื่อให้ฟังก์ชัน rl_search สามารถตรวจสอบได้ว่า ค่าในตัวแปร key มีอยู่ในตัวแปร array[ ] หรือไม่ (6 คะแนน)

```
int rl_search( int array[] , int key, int index, int size)
{
 if(____(1)_____)
  return -1;
 if(____(2)_____)
  return index;
 else
  return(rl_search(array, key,__(3)___, size) );
}
```

(1) D

(2) _____

(3) _____

9.2) จงเติมส่วนที่ขาดหายไป โดยเลือกจากกลุ่มของคำตอบที่กำหนดให้ เพื่อให้ฟังก์ชัน bSearch สามารถตรวจสอบได้ว่า ค่าในตัวแปร key มีอยู่ในตัวแปร array[ ] หรือไม่ (9 คะแนน)

```
int bSearch(int array[], int key, int size)
{  int low=0 , high= ___(1)___ , mid;
   while ( low<=high) {
      _____(2)_____;
      if (key< array[mid])
         high= mid-1;
      else if (key> array[mid])
         _____(3)_____;
      else   /* found match*/
         return _____(4)_____;
 }
 return -1;  /* no match*/
}
```

(1) J

(2) _____

(3) _____

(4) _____

ชื่อ _____ รหัส _____ section ___

Sorting (20 คะแนน)

10. เขียนผลลัพธ์ที่เปลี่ยนไปใน**แต่ละรอบ**ของการเรียงลำดับข้อมูลด้วย Selection sort และ Quick sort (16 คะแนน)

ตัวอย่างการแสดงผลลัพธ์ในแต่ละรอบ (ใช้ข้อมูลสำหรับตอบข้อ 10.3)

ข้อมูลเริ่มต้น: 3 5 2 7 4 2
รอบที่ 1: 3 2 5 4 2 7
รอบที่ 2: 2 3 4 2 5 7
รอบที่ 3: 2 3 2 4 5 7
รอบที่ 4: 2 2 3 4 5 7
หยุด

10.1) Selection sort (5 คะแนน)

ข้อมูลที่ต้องการเรียงคือ: 5 6 1 7 2 3

10.2) Quick sort (6 คะแนน)

กำหนดให้ Pivot เป็นข้อมูลทางขวามือสุด และ ให้การจบของแต่ละรอบคือการสลับค่า pivot ไปยังตำแหน่งที่ถูกต้อง

ข้อมูลที่ต้องการเรียงคือ: **5 6 9 1 7 2 3 8 10**

10.3) การเรียงลำดับข้อมูล: 3 5 2 7 4 2 ที่แสดงดังตัวอย่างข้างต้น ใช้วิธีการเรียงลำดับ (Sorting Algorithm) แบบใด (5 คะแนน)

---

11. จงเปรียบเทียบประสิทธิภาพ ในแง่ของเวลา (Best-case, Worst-case และ Average-case) ที่ใช้ในการทำงาน ในรูปแบบ O() ของ sorting algorithm ดังต่อไปนี้ (4 คะแนน)

Sorting Algorithm	Best Case	Average Case	Worse Case
Bubble sort	_____	$O(n^2)$	_____
Quick sort	_____	_____	$O(n^2)$
