PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Final Examination: Semester 1 **Academic Year:** 2009

Date: 4/10/2009 **Time:** 13.30 – 16.30

Subject Number: 241-207 **Rooms:** R300

Subject Title: Data Structure and Computer Programming Techniques

คำสั่ง : อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต : เครื่องเขียนต่าง ๆ เช่น ปากกา หรือ ดินสอ

ไม่อนุญาต : เครื่องคิดเลข, หนังสือ หรือ เอกสารใด ๆ เข้าห้องสอบ

เวลา : 3 ชั่วโมง (180 นาที)

รหัส น.ศ. .....

ชื่อ-นามสกุล .....

Section .....

ชื่อ ผู้สอน .....

คำแนะนำ

- ข้อสอบมีจำนวน **17** หน้า (ไม่รวมใบปะหน้า) คะแนนรวม **120** คะแนน
- เขียนคำตอบลงในข้อสอบ
- ใช้ดินสอทำข้อสอบได้ กรณีเขียนไม่ชัดหรืออ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- **อย่าลืม เขียน ชื่อ-นามสกุล รหัสนักศึกษา และ section ลงในข้อสอบทุกแผ่น**

ทุจริตในการสอบมีโทษขั้นต่ำปรับตกในรายวิชานี้ และพักการเรียน 1 ภาคการศึกษา

-- โทษสูงสุดคือ ไล่ออก –

สำหรับผู้ตรวจข้อสอบ

ตอนที่	คะแนน
1	
2	
3	
รวม	

รหัส น.ศ. ..... ชื่อ-นามสกุล .... section .....

ตอนที่ 1 ข้อสอบแบบตัวเลือก จงวงกลมล้อมรอบตัวเลือกในแต่ละข้อถูกต้องที่สุด เพียงข้อเดียว

(20 คะแนน)

1. จงเลือกประโยคคำสั่งที่ถูกต้องสำหรับ กำหนดตัวแปรชื่อ finput_ptr เพื่อชี้ไปยังชนิดข้อมูลแฟ้มข้อมูล (FILE)

ก. type finput_ptr as FILE;

ข. FILE * finput_ptr;

ค. finput_ptr FILE;

ง. FILE finput_ptr;

2. ประโยคคำสั่งในข้อใดเป็นประโยคคำสั่งที่ถูกต้องสำหรับ กำหนดตัวแปร finput_ptr สำหรับเปิดแฟ้มข้อมูล results.txt สำหรับเขียน

ก. finput_ptr = "results.txt" opened as "w";

ข. open finput_ptr as "results.txt" for "w";

ค. fopen(finput_ptr, "results.txt", "w");

ง. finput_ptr = fopen("results.txt", "w");

3. ตัวเลือกในข้อใดเป็นชุดคำสั่งที่ถูกต้องสำหรับใช้ทดสอบตัวแปรสำหรับชี้ไปยังแฟ้มข้อมูลชื่อ finput_ptr ว่าเปิดแฟ้มข้อมูลได้สำเร็จ โดยถ้าเปิดแฟ้มไม่สำเร็จให้พิมพ์ว่า Cannot open file.

ก.
```
if(finput_ptr == NULL ) {
        printf("Cannot open file.\n");
        exit(1);
}
```

ข.
```
if(finput_ptr != NULL ) {
        printf("Cannot open file. \n");
        exit(1);
}
```

ค.
```
while(finput_ptr != NULL ) {
        printf("Cannot open file. \n");
        exit(1);
}
```

ง.
```
while(finput_ptr = NULL ) {
        printf("Cannot open file. \n");
        exit(1);
}
```

4. จงเลือกคำสั่งที่ถูกต้องสำหรับปิดแฟ้มข้อมูล ของตัวแปรสำหรับชี้ไปยังแฟ้มข้อมูลชื่อ finput_ptr

ก. close finput_ptr;

ข. fclose(finput_ptr);

ค. fcloseall();

ง. finput_ptr (fclose);

5. ชุดคำสั่งในข้อใดที่ถูกต้องสำหรับอ่านข้อมูลมาหนึ่งแถวของแฟ้มข้อมูล ให้กับตัวแปรชื่อ line โดยจะอ่านค่าจนกว่าจะหมดแถว (เจอค่า \n) หรือ สิ้นสุดแฟ้มข้อมูล

ก.
```
int ch, count = 0;
fscanf( input_file , "%c", &ch);
while( (ch != '\n') && (ch != EOF) ) {
        line[count] = ch;
        count++;
        fscanf( input_file , "%c", &ch);
}
line[count] = NULL;
```

ข.
```
int ch, count = 0;
fread ( input_file , "%c", &ch);
while( (ch != '\n') && (ch != EOF) ) {
        line[count] = ch;
        count--;
        fread ( input_file , "%c", &ch);
}
line[count] = NULL;
```

ค.
```
int ch, count = 0;
fprintf( input_file , "%c", &ch);
while( (ch == '\n') && (ch == EOF) ) {
        line[count] = ch;
        count++;
        fprintf( input_file , "%c", &ch);
}
line[count] = -1;
```

ง.
```
int ch, count = 0;
fwrite( input_file , "%c", &ch);
while( (ch == '\n') && (ch == EOF) ) {
        line[count] = ch;
        count++;
        fwrite( input_file , "%c", &ch);
}
line[count] = -1;
```

6. เงื่อนไขที่ไม่จำเป็นต้องมีในการทำ binary search คือข้อใด

ก. ข้อมูลต้องเรียงลำดับ

ข. ต้องเข้าถึงกึ่งกลางของข้อมูลได้

ค. ต้องเพิ่ม และ ลบ ข้อมูลได้

ง. ถูกทุกข้อ

7. ชนิดข้อมูลใด มีความสัมพันธ์ของข้อมูลแบบตามลำดับ

ก. Strings

ข. Lists

ค. Queues

ง. ถูกทุกข้อ

จงตอบคำถาม ข้อ 8 ถึง 10 โดยใช้ข้อมูล ต่อไปนี้

ถ้ากำหนดให้มีการประกาศตัวแปร link list ตามโปรแกรมด้านล่าง ได้สำเร็จ

```
struct listnode {
 char data;
 struct listnode *nextptr;
};
typedef struct listnode LISTNODE;
typedef LISTNODE *LISTNODE_PTR;
```

```
void main() {
  int i;
  LISTNODE *list, *tail;
  list= malloc(sizeof(LISTNODE));
  list->data = 0;
  list->nextptr = malloc(sizeof(LISTNODE));
  list->nextptr->data = 1;
  list->nextptr->nextptr  = malloc(sizeof(LISTNODE));
  list->nextptr->nextptr->data = 2;
  tail=list;
}
```

8. ข้อใดที่เป็นคำสั่งที่ใช้เลื่อนตำแหน่งของ tail ไปยัง node ถัดไป

ก. tail++;

ข. tail = nextptr;

ค. tail += nextptr;

ง. tail = tail->nextptr;

9. ข้อใดเป็นเงื่อนไขสำหรับตรวจสอบว่าเป็น node สุดท้ายของ list

ก. (list == NULL)

ข. (list->nextptr == NULL)

ค. (list->data == NULL)

ง. ไม่มีข้อถูก

10. ข้อใดเป็นเงื่อนไขเพื่อตรวจว่า head และ tail เป็น nodeเดียวกัน ถ้ากำหนดให้ทั้งคู่ไม่เป็น NULL

ก. head == tail

ข. head->data == tail->data

ค. head->nextptr == tail->nextptr

ง. ไม่มีข้อถูก

11. เมื่อใส่ข้อมูลลง Stack ที่เต็มแล้ว จะเกิดสถานะที่เรียกว่า

ก. underflow

ข. overflow

ค. pop

ง. top

12. โครงสร้างข้อมูลแบบใดที่สามารถใส่ข้อมูลเข้าด้านหนึ่ง แล้วดึงข้อมูลออกอีกด้านหนึ่ง

ก. stack

ข. link list

ค. queue

ง. ถูกทุกข้อ

13. วิธีการวัดประสิทธิภาพของ algorithm สองอย่างได้แก่

ก. จำนวนวิธีการคำนวณ(calculation) และ จำนวนการอ่านหน่วยความจำ(read memory)

ข. ความซับซ้อนของโปรแกรม(complexity) และ จำนวนบรรทัดของโปรแกรม(code lines)

ค. เวลาที่ใช้(time) และ เนื้อที่ที่ต้องการเก็บข้อมูล(space)

ง. จำนวนข้อมูล(files) และ เนื้อที่ที่ใช้เก็บข้อมูล(space)

14. การเปรียบเทียบ algorithm เป็นค่า O() เปรียบเทียบจาก

ก. เวลาเป็น มิลลินาที

ข. จำนวนครั้งการเปรียบเทียบค่า key

ค. จำนวนบรรทัดของโปรแกรม

ง. ขนาดของโปรแกรม

15. กรณีใดไม่ใช้ สำหรับบอกประสิทธิภาพของ algorithm

ก. Best case

ข. Worst case

ค. Average case

ง. Null case

16. Worst case ในการทำ linear search เกิดขึ้นเมื่อ

ก. ข้อมูลที่ต้องการอยู่บริเวณกึ่งกลางของชุดข้อมูล

ข. ข้อมูลที่ต้องการไม่อยู่ในชุดข้อมูล

ค. ข้อมูลที่ต้องการอยู่บริเวณท้ายของชุดข้อมูล

ง. ข้อมูลที่ต้องการอยู่บริเวณท้ายของชุดข้อมูล หรือ ไม่อยู่ในชุดข้อมูล

17. ค่าประสิทธิภาพเฉลี่ย (Average case performance) ของ linear search คือ

ก. O(n)

ข. O(log n)

ค. O(n2)

ง. O(n log n)

18. ค่าประสิทธิภาพเฉลี่ย (Average case performance) ของ Bubble sort คือ

ก. O(n)

ข. O(log n)

ค. $O(n^2)$

ง. O(n log n)

19. การหาข้อมูลในชุดข้อมูลที่ต้องการเรียกว่า

ก. Traversal

ข. Search

ค. Sort

ง. Dequeue

20. ความสูงของ binary tree คำนวณได้จากจำนวน node n จากข้อใด

ก. $D_n = n \log_2 n$

ข. $D_n = n \log_2 n+1$

ค. $D_n = \log_2 n$

ง. $D_n = \log_2 n+1$

รหัส น.ศ. ..... ชื่อ-นามสกุล .... section .....

ตอนที่ 2 ข้อสอบแบบเติมคำ <u>จงเติมคำในช่องว่างให้ถูกต้องสมบูรณ์</u> (40 คะแนน)

1. จงเขียนประโยคคำสั่งภาษาซีต่อไปนี้ให้สมบูรณ์ (2 คะแนน)

เพื่อเปิดไฟล์ชื่อ data.txt เพื่อสร้างแฟ้มข้อมูลแบบตัวอักษรสำหรับเขียนข้อมูลชนิดตัวอักษร

Fptr = ……..(“data.txt”, “….”);

2. เราเรียกใช้ ฟังก์ชัน ..... เพื่อใช้เขียนข้อมูลแบบไบนารีให้กับแฟ้มข้อมูลที่ระบุ

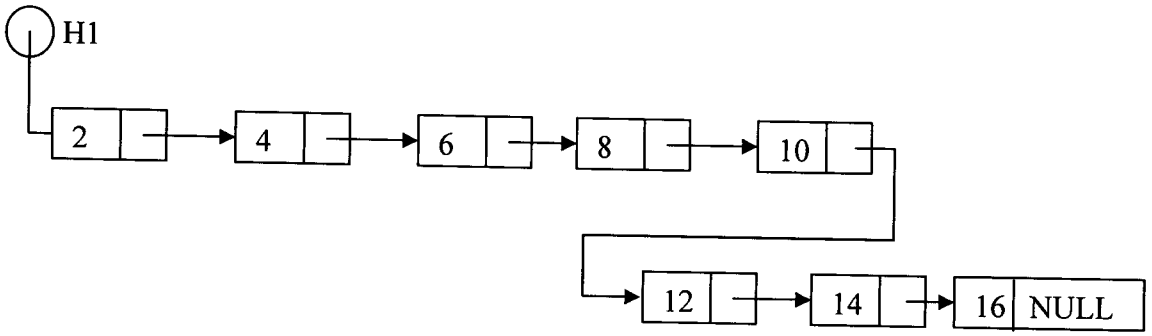
(2 คะแนน)

3. เราใช้ฟังก์ชัน ..... เพื่อเขียนข้อมูลลงไฟล์ตามรูปแบบข้อมูลที่กำหนดเองได้ (2 คะแนน)

4. ถ้ามี linked list ที่มีโครงสร้าง (5 คะแนน)

```
struct  node  {
        int  item;
        struct node * next;
};
```

ที่มีข้อมูลดังนี้



ตัวอย่าง List ที่ได้จากตัวชี้ H1 คือ

2 --> 4 --> 6 --> 8 --> 10 --> 12 --> 14 --> 16 --> NULL

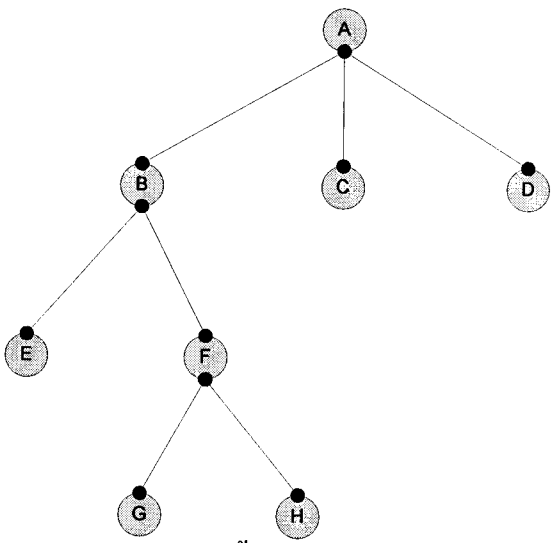
จงแสดง list ที่ได้จากตัวชี้ H2 เมื่อ กำหนดให้ H2 ได้มาจากส่วนของโปรแกรมต่อไปนี้

```
H3 = H1->next->next;
H2 = H1->next;
while ((H2 != NULL) && (H2->next != NULL)) {
        H1 = H1->next;
        H2 = H2->next->next;
}
H2 = H1->next;  /* Final H2 */
```

คำตอบ คือ

..... --> .....-->.....-->.....-->.....

5. หากกำหนดให้มี tree ดังต่อไปนี้ (5 คะแนน)



จงตอบคำถามต่อไปนี้

5.1 ความสูงของ tree คือ.....

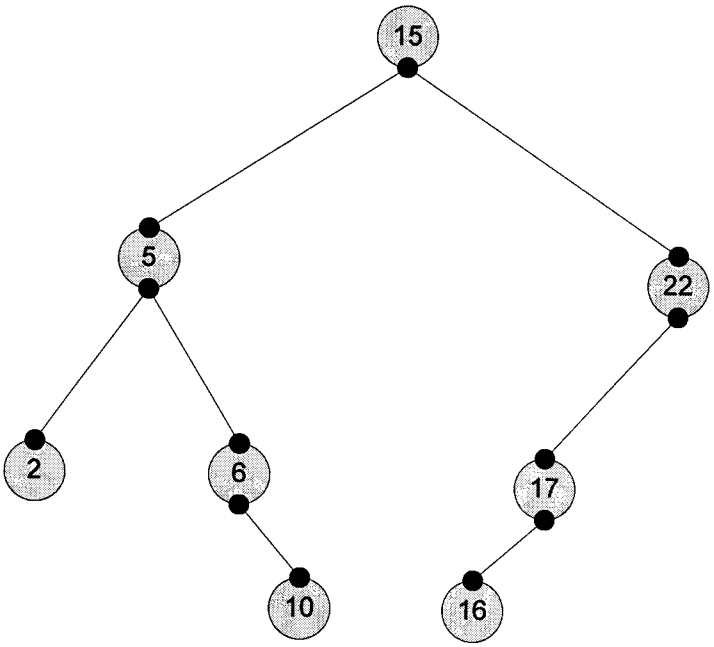
5.2 โหนด A มี degree เท่ากับ .....

5.3 Parent ของ โหนด F คือ .....

5.4 Leaf nodes ทั้งหมด ได้แก่ .....

5.5 Nodes ที่อยู่ใน level 1 ทั้งหมด ได้แก่ .....

6. จงหาผลลัพธ์ลำดับการเข้าถึง tree แบบต่าง ๆ ของ tree ต่อไปนี้ (24 คะแนน)



ตัวอย่าง เช่น

Inorder : 2 5 6 10 15 16 17 22

คำตอบ คือ

6.1 Preorder : ..... (8 คะแนน)

6.2 Postorder : ..... (8 คะแนน)

6.3 Breadth first : ..... (8 คะแนน)

รหัส น.ศ. ..... ชื่อ-นามสกุล .... section .....

ตอนที่ 3 ข้อสอบอธิบาย <u>จงอธิบายคำถามให้เข้าใจ หรือ เขียนโปรแกรมให้สมบูรณ์</u> (60 คะแนน)

1. จงเขียน ฟังก์ชัน ต่อไปนี้ให้สมบูรณ์เพื่อ แปลง อาเรย์ ให้เป็น linklist (12 คะแนน)

```
LISTNODE_PTR array2list(int *array, int n)
{
        LISTNODE *head, *tail;
        int i;

        head=malloc(.....);                          /*1.1 */    (2 คะแนน)
        head..... = array[0];                        /* 1.2 */    (2 คะแนน)

        tail = head;
        for(i=1; i<n; i++) {
                tail.....=malloc(.....); /* 1.3 */    (2 คะแนน)
                tail =.....;                         /* 1.4 */    (2 คะแนน)
                tail..... = array[i];                /* 1.5 */    (2 คะแนน)
        }
        return(.....);                               /* 1.6 */    (2 คะแนน)
}
```

2. กำหนดให้มี โปรแกรม ต่อไปนี้ (10 คะแนน)

```
#include <stdio.h>
#include <stdlib.h>

struct treenode {
  int data;
 struct treenode *leftptr;
 struct treenode *rightptr;
};
typedef struct treenode TREENODE;
typedef TREENODE *TREE;
```

```
int isempty(TREE t);
int value(TREE t);
void bstinsert(TREE *t,int value);
int printleaf(TREE t);
int count_leaf(TREE t);

int main()  {
  TREE t=NULL;

  bstinsert(&t,7);
  bstinsert(&t,8);
  bstinsert(&t,0);
  bstinsert(&t,6);
  bstinsert(&t,14);
  bstinsert(&t,1);
  /* binary tree is
   (7)
    /\
  (0) (8)
    \  \
   (6)  (14)
   /
(1)
  */
  printleaf(t);
  printf("Total leaf node(s) is %d\n", count_leaf(t));
  return 0;
}
```

```
void bstinsert(TREE *tp,int value)
{
        if(*tp==NULL) {
          *tp=malloc(sizeof(TREENODE));
          (*tp)->data=value;
          (*tp)->leftptr=NULL;
          (*tp)->rightptr=NULL;
        }
        else if(value<(*tp)->data)
                bstinsert(&((*tp)->leftptr),value);
        else if(value>(*tp)->data)
                bstinsert(&((*tp)->rightptr),value);
        else
            printf("Duplicate\n");

}
int printleaf(TREE t)
{
        if(t!=NULL)
        {
                if(t->leftptr ==NULL && t->rightptr ==NULL)
                        printf("%3d  ",t->data );
                printleaf(t->leftptr );
                printleaf(t->rightptr );
                return(1);
        }
        else
                return(0);
}
```

ซึ่งได้ผลลัพธ์การทำงานคือ

```
   1   14
Total leaf node(s) is 2
```

จงเขียนฟังก์ชัน count_leaf() ต่อไปนี้ให้สมบูรณ์ เพื่อ นับจำนวน leaf node ทั้งหมดของ binary tree แล้วส่งค่ากลับมาโดยตัวแปร count

*ข้อแนะนำ การใช้ static int จะสามารถนับจำนวนต่อเนื่องจากการเรียก ครั้งก่อนได้ เมื่อเรียกฟังก์ชันเดิมซ้ำ

```
int count_leaf(TREE t)  {
        static int count =0;
        if(t!=NULL)     {
                /* write your code here */
                   ……..
                   ……..
                   ……..
                   ……..
                   ……..
                   return(count);
        }
        else
                   return(0);
}
```

3. ถ้าต้องการเขียนโปรแกรมเพื่อแปลงเลขฐานสิบ เป็นเลขฐานสอง จงอธิบายว่าการใช้ push และ pop ของ stack ช่วยการแสดงผลลัพธ์ที่แปลงได้อย่างไร (3 คะแนน)

.....

.....

.....

.....

.....

4. ถ้ามีข้อมูลจำนวนมากที่ไม่ได้จัดเรียงในดิสก์ ควรใช้วิธีใดในการค้นหาข้อมูล แบบ external sort หรือ internal sort เพราะอะไร (3 คะแนน)

.....

.....

.....

.....

.....

5. กำหนดให้มีโปรแกรมจัดเรียงข้อมูลดังนี้ (8 คะแนน)

```
#include <stdio.h>
#define SIZE 8

void bubble_sort(char *a, int n);
void swap(char *data, int i, int j );

int main() {
 char data[SIZE] ="computer";
 int i;

 bubble_sort(data , SIZE );
 return 0;
}

void swap(char *v, int i, int j) {
   int temp;
   temp = v[i];
   v[i] = v[j];
   v[j] = temp;
}
```

จงเขียนฟังก์ชัน bubble_sort ต่อไปนี้ให้สมบูรณ์ เพื่อเรียงลำดับของ string ด้วยวิธี bubble sort เมื่อสมมุติให้มีข้อมูลใน string เป็นตัวอักษรตัวเล็ก a ถึง z เท่านั้น

```
void bubble_sort(char *a, int n)
{
   int i, j;

   for (.....)
      for (.....)
         if (.....)
            .....
            .....
}
```

6. จงอธิบายการทำงานของวิธีการจัดเรียงข้อมูลต่อไปนี้ และเขียนการเปลี่ยนแปลงของข้อมูลเมื่อมีการเรียงข้อมูลจากน้อยไปมากของข้อมูล 2 12 5 8 (12 คะแนน)

โดยวิธีต่อไปนี้ เช่น

Quick sort เป็นวิธีการเรียงโดยแบ่งส่วนข้อมูล แล้วจัดเรียงข้อมูลโดยเลือกคีย์มาสลับกับค่าน้อย

2 12 5 8

2 5 12 8

2 5 8 12

คำตอบ คือ

6.1 selection sort เป็นวิธีการเรียงโดย.....

.....

.....

.....

.....

.....

2 12 5 8

.....

.....

.....

.....

6.2 Bubble sort เป็นวิธีการเรียงโดย.....

.....

.....

.....

.....

.....

2 12 5 8

.....

.....

.....

.....

6.3 Insertion sort เป็นวิธีการเรียงโดย.....

.....

.....

.....

.....

.....

2 12 5 8

.....

.....

.....

.....

7. ถ้ากำหนดให้มี โครงสร้าง stack และ การเรียกใช้ต่อไปนี้ (12 คะแนน)

```
typedef struct node  {
   char       data;
   struct node* link;
} STACK_NODE;
int push     (STACK_NODE** pList, char  dataIn);
int pop      (STACK_NODE** pList, char* dataOut);
```

เมื่อต้องการใส่ข้อมูล 'C' ให้ใช้

```
Push(pList, 'C');
```

และเมื่อนำข้อมูล 'C' ออกมาโดยใช้

```
Pop(plist, ch);
printf("%c", ch);
```

เมื่อ pList เป็นพอยน์เตอร์ของ STACK_NODE และ ch เป็น char

จงเขียนคำสั่งการประมวลผล stack ต่อไปนี้ให้สมบูรณ์ เพื่อใส่ข้อมูล 'CoE' ลงใน stack และ แสดงการนำข้อมูล 'CoE' ออกจาก stack ตามลำดับข้อมูลโดยไม่ใช้ loop

```
int main (void) {
        STACK_NODE* pStackTop;
        pStackTop = NULL;
        ……..
        ……..
        ……..
        ……..
        ……..
        ……..
        ……..
        ……..
        ……..
        return 0;
}
```