CS 2412 FINAL EXAMINATION

December 5th, 2007

Duration: 3 hours

Student Name:

Student Number:

Note: This exam set includes 10 problems in 12 pages. Please check the pages before you solve problems.

Problem 1. (10 marks)

What is the output of the following segment of a program?

Segment 1.

```
 ......
 typedef struct stackentry{
  int info;
  struct stackentry *next;
} StackEntry;
typedef StackEntry QueueNode;
......
void CreateStack(Stack *s);
void Push(StackEntry item,Stack *s);
void Pop(Stackentry *item,Stack *s);
void CreateQueue(Queue *q);
void AppendNode(QueueNode *p,Queue *q);
void ServeNode(QueueNode **p,Queue *q);
......
StackEntry *EN;
QueueNode NO;
......
for(i=0;i<10;i++) INP[i]=i;
CreateStack(St);
CreateQueue(Qu);
for(i=0;i<10;i++) {
 NO.info=INP[i];
 NO.next=NULL;
 Push(NO,St);
}
for(i=0;i<10;i+=2){
 Pop(EN,St);
 AppendNode(EN,Qu);
}
for(i=0;i<5;i++) {
 Pop(EN,EN,St);
 printf("%d",EN->info);
 ServeNode(&EN,Qu);
 printf("%d",EN->info);
}
```

The output is:

Segment 2.

```
......
#define GT(a,b) ((a)>(b))
typedef struct lentry{
  int key;
  char info;
} ListEntry;
struct list{
  ListEntry entry[100];
  int count;
};
int i,bottom,top,middle,target;
......
for(i=0;i<50;i++){
  list.entry[i]=2*i;
  list.count++;
}
bottom=0;
top=list.count-1;
middle=-1;
target=20;
while(bottom<top){
  middle=(top+bottom)/2;
  if(GT(target,list.entry[middle].key))
    bottom=middle+1;
   else
    top=middle;
 }
printf("entry %d 's key is %d\n",top,list.entry[top].key);
```

The output is:

Problem 2. (5 marks)

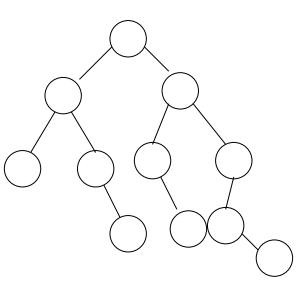
Suppose we have established a binary search tree of positive integers. The main definitions are as follows.

```
typedef int TreeEntry;
typedef struct treenode{
  TreeEntree entry;
  treenode *left;
  treenode *right;
} TreeNode;
TreeNode *root;
```

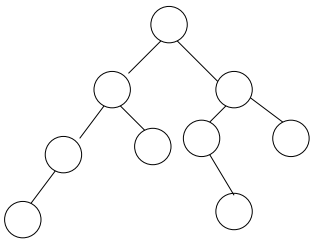
Write a C recursion function to count how many even numbers are in the tree.

Problem 3. (5 marks)

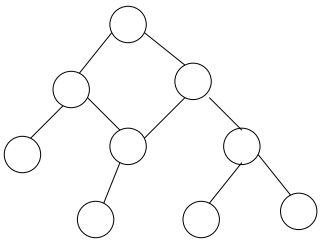
Investigate the following 3 graphs A, B and C:



A



B



C

Indicate which are:

(1) trees:

(2) binary trees:

(3) AVL trees:

(Note: a graph might belong to more than one type)

Problem 4. (10 marks)

The inorder and postorder traversals of a binary tree with English letters are:

DBGEAHFIC
DGEBHIFCA

respectively.

(1) Draw the binary tree.

(2) Write down the preorder traversal of this tree.

Problem 5. (10 marks)

Suppose a sequence of numbers is given as:

$$8, 20, 40, 14, 13, 35, 45, 50, 42, 41$$

(1) Draw a binary search tree by inserting the above numbers.

(2) Draw an AVL tree by inserting the above sequence.

(3) When you search a number from the sequence, using which tree is more efficient in average? Explain why.

Problem 6. (10 marks)

Suppose a sequence of integers is as follows:

$$23, 16, 15, 54, 32, 20, 11, 2, 3, 66, 55, 25, 34, 5, 63, 65, 22, 4, 1, 100$$

(1). Draw a B-tree of order 5 to store these integers.

(2). Draw a trie to store these integers. Since the digits of the numbers are $0, 1, 2, 3, 4, 5, 6$, the trie has at most 7 branches..

Problem 7. (10 marks)

Draw the radix sort step by step for the following 14 names:

`Tim Dot Eva Roy Tom Kim Guy Amy Jon Ann Jim Kay Ron Jan`

Problem 8. (10 marks)

Suppose the adjacency table of a directed graph with weights is:

	0	1	2	3	4	5
0	0	2	∞	8	∞	∞
1	∞	0	1	2	∞	8
2	∞	∞	0	∞	6	∞
3	∞	3	∞	0	∞	4
4	∞	∞	∞	∞	0	∞
5	3	∞	∞	∞	∞	0

(1) Draw the directed tree.

(2) Find out shortest paths from vertex 0 to other vertices using the greedy algorithm. Show the steps of how you find out the paths.

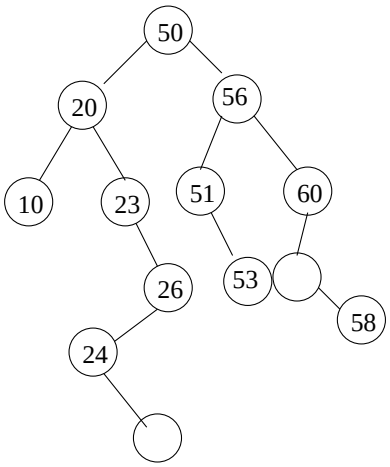
Problem 9. (15 marks)

Suppose we have the following definition of B-tree and related C codes. Now you are asked to write a C function which first finds out two given keys (two entries), then exchanges the Data stored in these two entries.

```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 4
#define MIN 2
typedef int Key;
typedef struct treenode Treenode;
typedef struct treeentry Treeentry;
struct treeentry{
  Key key;
  int Data;
};
struct treenode {
  int count;
  Treeentry entry[MAX+1];
  Treenode *branch[MAX+1];
};
Treenode *SearchTree(Key target,Treenode *root,int *targetpos);/* the function
return value points to the node containing target in position targetpos*/
```

Problem 10. (15 marks)

(1) First fill two missing numbers in the following binary search tree. Then delete nodes with numbers 23, 56 from the tree. Draw the resulting tree.



(2) The following is a B-tree of order 5. Delete entries p and d from the tree and draw the resulting B-tree.