國立成功大學
111學年度碩士班招生考試試題

編　號：201

系　所：電機資訊學院-資訊聯招

科　目：程式設計

日　期：0219

節　次：第 2 節

備　註：不可使用計算機

※ 考生請注意：本試題不可使用計算機。 請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

Part I. 資料結構 (50%)

Data Structures: **There are 12 problem sets in the paper. You shall provide a "single" table in "1-page" as follows to summarize your answers.**

(1) [2%, 2%]	To-Do A: To-Do B:
(2) [2%]	
(3) [2%, 3%]	(i) (ii)
(4) [4%]	
(5) [2%]	
(6) [2%, 3%]	(i) (ii)
(7) [2%]	
(8) [4%]	(A) (B) (C) (D) (E)
(9) [1%, 1%, 1%, 1%]	(A) (B) (C) (D)
(10) [2%, 2%]	(i) (ii)
(11) [2%, 2%, 2%]	(i) (ii) (iii)
(12) [2%, 2%, 2%, 2%]	(i) (ii) (iii) (iv)

請於答案卷上作答，在此作答不計分

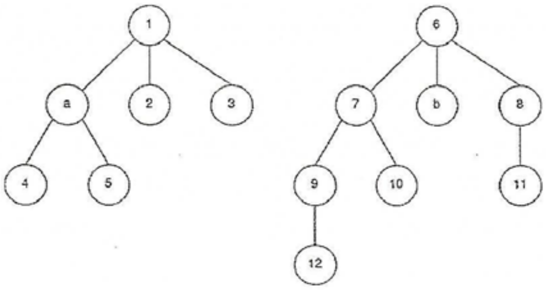
1) **[4%]** The following C program converts a preorder expression to its postorder form. For example, "+AB++ABC" is converted to "AB+AB+C+" with the program. Please fill in the missing parts of the program in C language.

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

#define MAX_INPUT 100000

typedef struct _Stack_node{
    char val;
    struct _Stack_node *prev;
} Stack_node;

typedef struct _Stack{
    Stack_node *top;
} Stack;

Stack *new_stack(){
    Stack *stk = malloc(sizeof(Stack));
    stk->top = NULL;
    return stk;
}

char Stack_pop(Stack *stk){
    char value = stk->top->val;
    Stack_node *tmp = stk->top;
    /** TODO (A) **/
    free(tmp);
    return value;
}

void Stack_push(Stack *stk, char value){
    Stack_node *new_node = malloc(sizeof(Stack_node));
    new_node->val = value;
    /** TODO (B) **/
    stk->top = new_node;
```

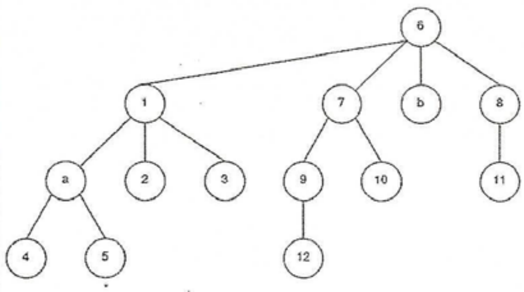
```
        return;
    }

    int Stack_is_empty(Stack *stk){
        return stk->top == NULL;
    }

    void pre_to_post(char *);
    void post_to_pre(char *);

    int main(){
        char input[MAX_INPUT];
        scanf("%s", input);
        pre_to_post(input);
        return 0;
    }

    void pre_to_post(char *input){
        Stack *stk = new_stack();
        for (int i = 0; input[i] != '\0'; ++i){
            switch(input[i]){
                case '+':
                case '-':
                case '*':
                case '/':
                case '^':
                case '%':
                    Stack_push(stk, input[i]);
                    break;
                default:
                    printf("%c", input[i]);
                    while (!Stack_is_empty(stk) && stk->top->val == 'A'){
                        char top = Stack_pop(stk);
                        printf("%c", Stack_pop(stk));
                    }
                    if (!Stack_is_empty(stk)) Stack_push(stk, 'A');
            }
```

```
        }
        printf("\n");
    }
```

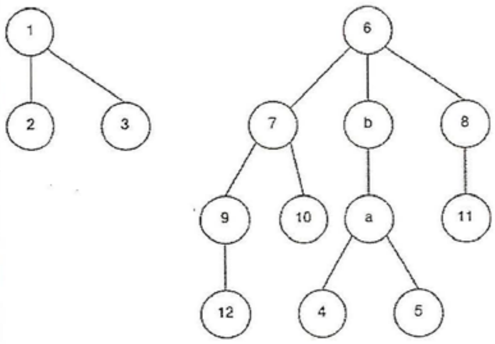
2) **[2%]** We rely on the tree data structure to represent a set of elements. Two elements are in the same set if they share the common root. Given two sets as follows, which is true if we union the two sets? Assume the union follows the height union rule.



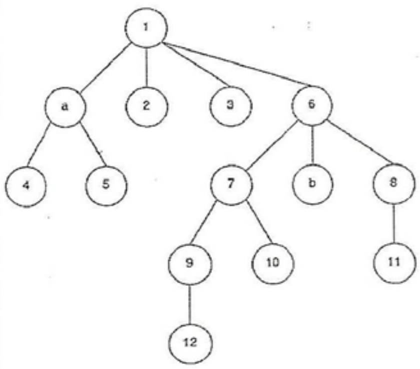
(A)



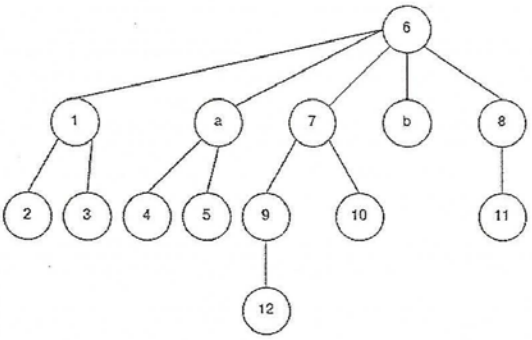
(B)



(C)

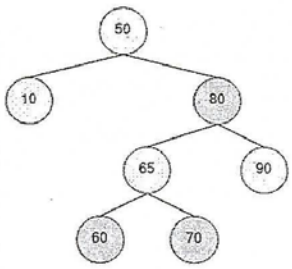


(D)

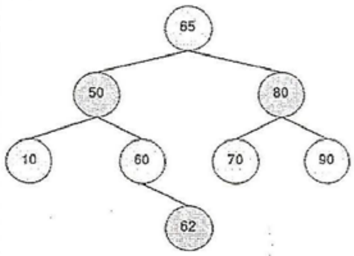


(E) None.

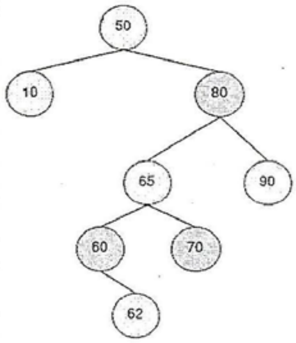
3) **[2%, 3%]** You are given the red-black tree in the following, where the dark colored nodes are red nodes. We insert the data item "62" into the tree. (i) Which of the followings is true? (ii) Please illustrate its 2-3-4 tree counterpart representation after 62 is inserted.



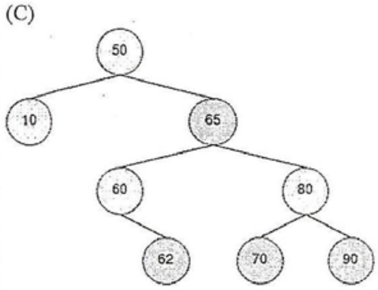
(A)



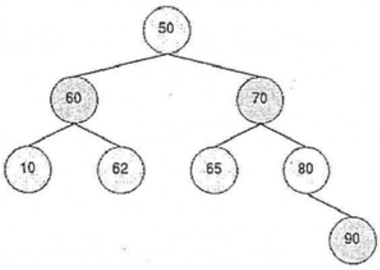
(B)



(C)



(D)



4) **[4%]** You are given a Bloom filter with the three hash functions as follows, where *modf*(.) returns the fractional part of a floating point number.

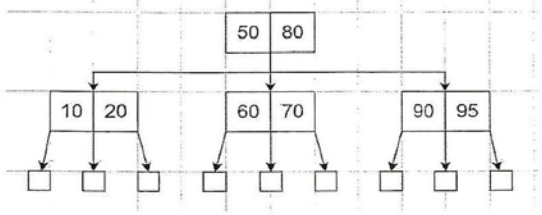
$$f_1(x) = floor(modf(x * 0.31) * 16)$$
$$f_2(x) = floor(modf(x * 0.24) * 16)$$
$$f_3(x) = floor(modf(x * 0.13) * 16)$$

Bloom Filter																
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	0	1	1	0	0	1	0	0	0	1	0	0	1	1	0	0

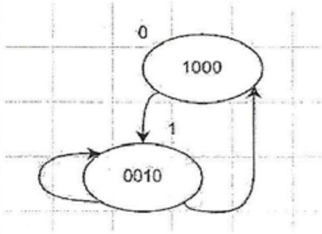
Which of the followings is (are) true?

(A) 8 is not in
(B) 8 may be in
(C) 8 is in
(D) 9 is not in
(E) 9 may be in
(F) 9 is in

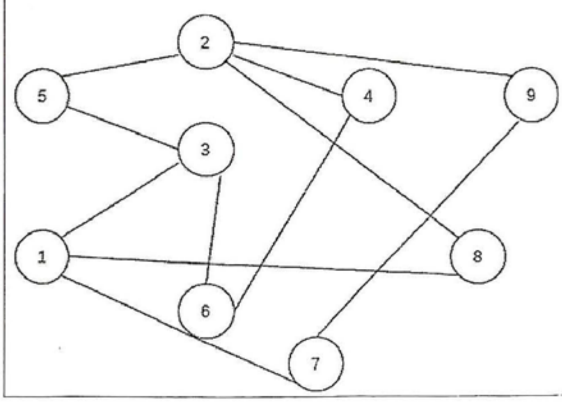
5) **[2%]** A B-tree is shown in the following. We remove the data items 70, 10, 60 and 95 in order over the B-tree. What is the resultant of B-tree after the deletions?



6) **[2%, 3%]** You are given a Patricia as follows. We insert 1001, 1100, 0000 and 0001 in order into the Patricia. (i) Please show the resultant Patricia after the four data items are inserted. (ii) Please also depict its compressed binary trie counterpart.



7) **[2%]** Consider the graph as follows. The DFS traversal is performed, starting from node 1. What are the potential sequences for the traversal?



(A) 1 7 9 2 4 6 3 5 8
(B) 1 3 6 4 2 5 9 7 8
(C) 1 7 8 3 5 6 4 2 9
(D) 1 8 2 4 6 3 5 9 7
(E) 1 2 5 6 4 2 9 7 8

8) **[4%]** The following is a hash table with linear probing. Assume the hash function is f(x) = x mod 17 for indexing the hash table. Data items 23, 52, 11, 1, 50, 99, 65, 20, 35, 34 and 82 are inserted into the hash table in order. Please fill the five blanks in the table.

key	value
0	34
1	(A) ?
2	(B) ?
3	(C) ?
4	(D) ?
5	(E) ?
6	23
7	
8	
9	
10	
11	11
12	
13	
14	99
15	65
16	50

9) **[1%, 1%, 1%, 1%]** The following C program implements Quick sort. Please fill the four missing parts of the program.

```
int Partition(int *arr, int front, int end){
    int pivot = arr[end];
    int i = front -1;
    for (int j = front; j < end; j++) {
        if (arr[j] < pivot) {
```

```
            _(A)_;
            swap(&arr[i], &arr[j]);
        }
    }
    i++;
    swap(&arr[i], &arr[end]);
    _(B)_;
}
void QuickSort(int *arr, int front, int end){
    if (front < end) {
        int pivot = Partition(arr, front, end);
        QuickSort(___C___);
        QuickSort(___D___);
    }
}
```

10) **[2%, 2%]** The following data structure represents a binary tree, and it contains a function named "unknown".

```
struct node{
    int data;
    struct node *left, *right;
};

void unknown(struct node *p) {
    struct node *q;
    if (p -> left != NULL) unknown(p->left);
    if (p -> right != NULL) unknown(p->right);
    q=p->left;
    p->left=p->right;
    p->right=q;
}
```

(i) Which of the order does the function perform?
(A) preorder
(B) inorder
(C) postorder

(D) level order

(E) None of the above.

(ii) You are given the following input. Once the unknown(r) is performed, what is the value of the rightmost leaf node?

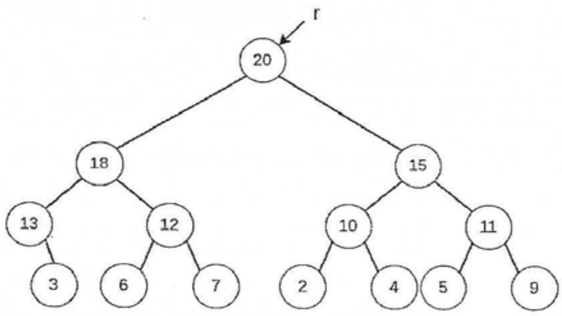
(A) 7

(B) 6

(C) 3

(D) 2

(E) None of the above



11) **[2%, 2%, 2%]** A job priority queue is implemented using a Min-Heap in which a lower key value represents a higher priority. The jobs are entered and stored in the Min-Heap as shown in the following array Q.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
key value	--	6	8	10	12	24	15	13	20	18	26						

(i) [Step 1] Next job is extracted from the job queue for execution. What is the value of $Q[4]$ in the remaining job queue?

(A) 15

(B) 18

(C) 20

(D) 26

(E) None of the above

(ii) [Step 2] After Step (1) is executed, next job is extracted from the job queue for execution. What is the value of $Q[5]$ in the remaining job queue?

(A) 13

(B) 15

(C) 18

(D) 24

(E) None of the above

(iii) [Step 3] After step (2) is executed, a new job with priority 11 is inserted in to the job queue. What is the value of $Q[9]$ in the remaining job queue?

(A) 18

(B) 20

(C) 24

(D) 26

(E) None of the above

12) **[2%, 2%, 2%, 2%]** Consider the two missing parts for the decrease key operation in F-heap.

```
DecreaseKey(H, x, k){
  if k > x.key return "error"
  x.key = k
  y = x.p    //note: x.p represents the parent node of x
  if y ≠ NIL and (1) {
     CUT(H, x, y)
     CASCADING − CUT(H, y)
  }
  if x.key < H.min.key
      H.min = x
}

CUT(H, x, y){
  x.p = (2)
  x.mark = FALSE
}
```

```
CASCADING − CUT(H, y){
 z = y.p
 if z ≠ NIL
     if y.mark == FALSE
          y.mark = TRUE
     else {
          CUT(H, y, z)
          CASCADING − CUT(H, z)
  }
}
```

(i) Which of the following for (1) is true?

(A) $x.key < y.key$
(B) $x.key > y.key$
(C) $x.key \geq y.key$
(D) $x.key \leq y.key$

(ii) Which of the followings is true for (2)?

(A) NIL
(B) y
(C) x
(D) None of the above

(iii) What is the amortized time complexity of function <u>DecreaseKey</u>?

(A) $O(1)$
(B) $O(n)$
(C) $O(lg\ n)$
(D) $O(lg\ n^2)$

(iv) What is the time complexity of inserting node into F-heap?

(A) $O(1)$

(B) $O(n)$

(C) $O(lg\ n)$

(D) $O(lg\ n^2)$

Part II. 演算法 (50%)

13) (10%) Give asymptotic tight bound (Θ) for $T(n)$ where $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}\log(n)$. (Assume that $T(n)$ is a constant for sufficiently small n.)

14) (10%) Given a sequence $K = \langle k_1, k_2, k_3, k_4, k_5 \rangle$ of five distinct keys in sorted order (so that $k_1 < k_2 < k_3 < k_4 < k_5$) and six dummy keys $d_0, d_1, d_2, d_3, d_4, d_5$ representing values not in K, we have a probability p_i for k_i and a probability q_i for d_i. Determine the cost of an optimal binary search tree for K with the following probabilities:

i	0	1	2	3	4	5
p_i		0.08	0.15	0.05	0.1	0.12
q_i	0.04	0.1	0.08	0.1	0.06	0.12

15) (10%) Given a directed and strongly connected graph $G = (V, E)$, please design an algorithm to determine whether it contains a (directed) cycle of odd length.

16) (10%) Find a feasible solution or determine that no feasible solution exists for the following system of difference constraints:

$x_1 - x_3 \le 1$
$x_2 - x_3 \le -4$
$x_4 - x_5 \le 2$
$x_3 - x_4 \le 7$
$x_5 - x_1 \le 5$
$x_4 - x_2 \le 10$
$x_1 - x_2 \le 2$
$x_5 - x_3 \le -1$

編號： 201

國立成功大學 111 學年度碩士班招生考試試題

系　所：電機資訊學院-資訊聯招
考試科目：程式設計
考試日期：0219，節次：2

17) (10%) Given a finite set X and a family F of subsets of X such that every element of X belongs to at least one subset in F, the set-covering problem is to find a minimum-size subset $\zeta \subseteq F$ whose members cover all of X: $X = \bigcup_{S \in \zeta} S$. The following algorithm is a δ-approximation algorithm to find a set cover ζ satisfying $|\zeta| \leq \delta|\zeta^*|$ where ζ^* is an optimal solution. Please compute the minimum value of δ.

```
GREEDY-SET-COVER(X, F)
U = X
ζ = ∅
while U ≠ ∅
    select an S ∈ F that maximizes |S ∩ U|
    U = U − S
    ζ = ζ ∪ {S}
return ζ
```