

國立中正大學

111 學年度碩士班招生考試

試題

[第 2 節]

科目名稱	資料結構
系所組別	資訊管理學系-乙組

—作答注意事項—

※作答前請先核對「試題」、「試卷」與「准考證」之系所組別、科目名稱是否相符。

1. 預備鈴響時即可入場，但至考試開始鈴響前，不得翻閱試題，並不得書寫、畫記、作答。
2. 考試開始鈴響時，即可開始作答；考試結束鈴響畢，應即停止作答。
3. 入場後於考試開始 40 分鐘內不得離場。
4. 全部答題均須在試卷（答案卷）作答區內完成。
5. 試卷作答限用藍色或黑色筆（含鉛筆）書寫。
6. 試題須隨試卷繳還。

國立中正大學 111 學年度碩士班招生考試試題

科目名稱：資料結構

本科目共 2 頁 第 1 頁

系所組別：資訊管理學系-乙組

1. If n is 6, then the variable space requirement of this program is? (6 pt)

```
float rsum(float list[ ], int n) {  
    if (n) return rsum(list, n-1) + list[n-1];  
    return 0;  
}
```

2. Calculate the total steps of the following function (6 pt)

```
void add( int a[ ][ ], int b[ ][ ], int c[ ][ ], int m, int n) {  
    for (int i = 0; i < n; i++)  
        for (int j = 0; j < m; j++)  
            c[i][j] = a[i][j] + b[i][j];  
}
```

3. Please transform the expressions from postfix to prefix. (12 pt)

(A) $ab/c-de*+ac*-$

(B) $abc+*d/g-$

4. Which(s) is/are true in the following descriptions? (24 pt)

(A) In polynomial addition, the number of coefficient additions is at least 1.

(B) In polynomial addition, the number of creation of new nodes is at most $m+n-1$.

(C) In polynomial addition, the number of creation of new nodes is at least $\max\{m,n\}$.

(D) In polynomial addition, the number of coefficient additions is at most $\min\{m,n\}$.

(E) In polynomial addition, the number of exponent comparisons is at most $m+n$.

(F) $\&a \rightarrow val$ means $(\&a) \rightarrow val$

5. Please complete the code of "Add to a circular queue". (12 pt)

```
void addq(element item)  
{  
    /* hint: front, rear, MAX_QUEUE_SIZE, queue[] */  
    .....  
    if (.....)  
        queueFull();  
    .....  
}
```

6. Please complete the code of "Invert single linked list". (20 pt)

```
typedef struct list_node *list_pointer;  
typedef struct list_node {
```

國立中正大學 111 學年度碩士班招生考試試題

科目名稱：資料結構
系所組別：資訊管理學系-乙組

本科目共 2 頁 第 2 頁

```

char data;
list_pointer link;
};

list_pointer invert(list_pointer lead)
{
    list_pointer middle, trail;
    middle = NULL;
    while (.....) {
        .....
        .....
        .....
        .....
    }
    return middle;
}

```

7. Write out the following results? (20 pt)

<pre> a[0] 6 6 8 1 0 0 15 2 0 3 22 3 0 5 -15 4 1 1 11 5 1 2 3 6 2 3 -6 7 4 0 91 8 5 2 28 </pre>	<pre> 7(A) void transpose(term a[], term b[]) { int n, i, j, currentb; n = a[0].value; b[0].row = a[0].col; b[0].col = a[0].row; b[0].value = n; if (n > 0) { currentb = 1; for (i = 0; i < n; i++) for (j = 1; j <= n; j++) if (a[j].col == i) { b[currentb].row = a[j].col; b[currentb].col = a[j].row; b[currentb].value = a[j].value; currentb++; } } } </pre>	<pre> 7(B) void fast_transpose(term a[], term b[]) { int row_terms[MAX_COL], starting_pos[MAX_COL]; int i, j, num_cols = a[0].col, num_terms = a[0].value; b[0].row = num_cols; b[0].col = a[0].row; b[0].value = num_terms; if (num_terms > 0) { for (i = 0; i < num_cols; i++) row_terms[i] = 0; for (i = 1; i <= num_terms; i++) row_term [a[i].col]++; starting_pos[0] = 1; for (i = 1; i < num_cols; i++) starting_pos[i] = starting_pos[i-1] + row_terms [i-1]; for (i = 1; i <= num_terms; i++) { j = starting_pos[a[i].col]++; b[j].row = a[i].col; b[j].col = a[i].row; b[j].value = a[i].value; } } } </pre>
---	---	---