中原大學 106 學年度 ☑上學期 考試題卷 ☑期中 68

科目名稱：資料結構 開課班級：資工二甲/資工二乙 油印份數：70/61

考試時間：11 月 7 日 第 4 節 科目代號：CS103D/CS103E 教師姓名：吳宜鴻

☑不可使用計算機、翻譯機或手機 ☑可以使用紙本字典 本份試題共 4 頁，本版面為第 1 頁

☑直接在命題紙上作答（背面可作為計算用紙） ☑評分將不予部分給分，作答務必力求完整

系級：資工二乙 學號：10527237 姓名：孫湘屏

I. Single-Choice Problems (50%) 每題 3 分，共 20 題，答錯一題倒扣 1 分，滿分以 50 分為上限 +50

A 1. Which of the following is the benefit of *modularity*? (A) it isolates errors (B) it runs faster (C) it is easier to make a copy (D) it saves more space

C 2. Which of the following statements is FALSE about a *constructor* in C++? (A) a constructor has no return type (B) a constructor may have a parameter (C) a class has only one constructor (D) a constructor has the same name as the class

D 3. What happens if a recursive function never reaches a *base case*? (A) the function returns the correct value (B) the function returns an incorrect value (C) the function terminates immediately (D) an infinite sequence of recursive calls occurs

A 4. In the recursive solution to the *Eight Queens* problem, the problem size decreases by ___ at each recursive step. (A) one column (B) two columns (C) one square (D) two squares

B 5. The *midpoint* of a sorted array has the index _____, where x is the index of the first item in the array, and y is the index of the last item in the array. (A) $x/2 + y/2$ (B) $x + (y - x)/2$ (C) $(y - x)/2$ (D) $(x - y)/2$

C 6. For an *array* containing 2, 3, 5, 9, 13, 16, and 19, what value does a recursive binary search algorithm return when it searches for 9? (A) -1 (B) 1 (C) 3 (D) 6

B 7. Which of the following is NOT true about converting *infix* expressions to *prefix* expressions? (A) an operator will move "to the left" with respect to the operands (B) an operator will move "to the right" with respect to the operands (C) the operands always stay in the same order with respect to one another (D) all parentheses are removed

C 8. If a *stack* is used to check balanced braces, the stack _____ when the end of the string is reached? (A) has one "{" (B) has one "}" (C) is empty (D) has one "{" and one "}"

B 9. A pointer-based stack MUST define an _____. (A) explicit copy constructor (B) explicit destructor (C) integer variable to keep the maximum size of the stack (D) integer variable to keep the top of the stack

D 10. Given the declaration: Node *pre = NULL, *cur= head; which of the following statements can be put into a loop to traverse a *linked list* pointed to by head? (A) cur = pre; pre = cur->next; (B) cur = pre->next; pre = cur; (C) cur = pre; pre = cur->next; (D) pre = cur; cur = pre->next;

A 11. Which of the following functions is NOT about *file processing*? (A) free (B) fread (C) fseek (D) fopen

科目名稱：資料結構　　開課班級：資工二甲/資工二乙

考試時間：11 月 7 日 第 4 節　　科目代號：CS103D/CS103E

D 12. The last node of a circular linked list _____. (A) has the value NULL (B) has a next pointer whose value is NULL (C) cannot store any data (D) has a next pointer that points to the first node of the list

B 13. Which of the following is TRUE about a *destructor* in C++? (A) a class can have several destructors (B) the compiler will generate a destructor if the programmer does not provide one (C) a programmer must provide a destructor for every class (D) a destructor destroys all instances of a class

A 14. A(n) _____ is a C++ construct that enables a programmer to define a new data type. (A) class (B) interface (C) variables (D) object

D 15. A function can indicate that an error has occurred by throwing an _____. (A) interface (B) implementation (C) inheritance (D) exception

D 16. In the recursive solution to the *Towers of Hanoi* problem, the number of disks to move _____ at each recursive call. (A) increases by half (B) decreases by half (C) increases by one (D) decreases by one

B 17. What is fundamentally wrong with computing the *Fibonacci* sequence recursively? (A) it has two base cases (B) it eventually converges to a particular value (C) each call to the function results in at most two recursive calls (D) it is an instance of binary recursion

A 18. The base case for the recursive solution to the *Towers of Hanoi* problem is when a tower has _____. (A) only one disk (B) no disk (C) all disks (D) exactly two disks

C 19. Which of the following strings is NOT in the *language* defined by the following *grammar*? (A) 0a1 (B) 00b1 (C) 00b11 (D) 00a11

<X> = a | 0<Y>

<Y> = b | <X>1

0<Y>
0 <X> 1
0 0<Y> 1
0 0<X> 1 1

C 20. Among the following statements about *recursive binary search*, which one is TRUE? (A) it can be applied to an unsorted array (B) it starts by comparing with the first item in the array (C) it has two base cases (D) it searches exactly one half of the array

<U><S>
<U><U><B>
<U>

+15

II. Simple-Answering Problems (30%) 每格 3 分，共 11 格，作答完整才得分，滿分以 30 分為上限

1. Consider the language **S** as defined by the following *grammar* to answer the following questions.

<S> = <D> | <U> <S>
<D> = 1 | 0
<U> = A | B

(i) Write all strings in the language that have exactly three characters.

Answer: **(1)** AA1, AA0, AB1, AB0, BA1, BA0, BB1, BB0

(ii) Is the string AABB01 included in this language? Why?

Answer: **(2)** No, 最後不可能有兩個數字，只能一個

(iii) Write the grammar to define a language of bit-strings (only contain 0 and 1) so that the first character must be 1 and the last character must be 0.

Answer:

<S> = 10 | <D>0 | <D>1

<D> = <1><S><0>

(3)

II. Simple-Answering Problems (30%) 每格 3 分，共 11 格，作答完整才得分，滿分以 30 分為上限

2. Fill the blanks to complete the pseudo code about the recursive solution of the *Towers of Hanoi* problem.

```
SolveToH( countOfDisks, pA, pB, pC )
    if (countOfDisks == 1)
        MoveOneDisk(pA, pC)          // Move one disk from pA to pC
    else
    {   SolveToH(countOfDisks - 1, (4) pA, pB, pC )
        SolveToH(1, pA, pB, pC)
        SolveToH(countOfDisks - 1, (5) pB, pC, pA )
    }
```

*+9/-10*6+2 8+7 3 4

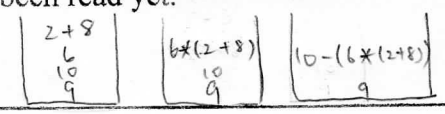
3. Given the *infix* expression **(9 + ((10 - (6 * (2 + 8))) / (7 + 3))) * 4**, answer the following questions about evaluating it by using a *stack*.

(i) The first step transforms it into the postfix expression:

Postfix expression: **(6)** 9 10 6 2 8 + * 7 3 + / - + 4 *

(ii) The second step is to calculate its value by using a stack. Draw the content of stack at the moment before the operator '/' has NOT been read yet.

Content of stack at the moment: **(7)**

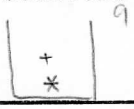


(iii) The above way to evaluate postfix expression can also be applied to prefix expression. The first step transforms it into the prefix expression:

Prefix expression: **(8)** * + 9 / - 10 * 6 + 2 8 + 7 3 4

(iv) The second step is to calculate its value by using a stack. Draw the content of stack at the moment before the operator '/' has NOT been read yet.

Content of stack at the moment: **(9)**



4. Given the *array*-based implementation of ADT **List**, the following function can reverse the order of the items stored in **aList**. Assume that there are 5 items in **aList**.

A B C D E
B C D E A
C D E B A

5×4 + 1+2+3

```
reverseList(aList: List)
    for (i = 1 to aList.getLength() - 1)
    {   aList.retrieve(1, dataItem, success);
        aList.insert(aList.getLength() – i + 2, dataItem, success);    // the operation insert
        aList.remove(1, success);    // the operation remove
    }
```

4×4 + 6

(i) What is the total number of data movements due to the two operations **insert** and **remove**?

Answer: **(10)** >6

(ii) If the order of **insert** and **remove** are exchanged, what is the total number of data movements?

Answer: **(11)** 22

科目名稱：資料結構　　開課班級：資工二甲/資工二乙

考試時間：11 月 7 日 第 4 節　　科目代號：CS103D/CS103E

III. Advanced Problems (20%)**每一空格 3 分，共 7 空格，作答完整才得分，滿分以 20 分為上限**

1. Give short C/C++ codes as an example to explain each term. ***Get no point if there is no code!***

- **Call by value**:

```
int A(int x){
(1) };
```

- **Call by address**:

```
int A(int &x){
(2) };
```

- **Call by reference**:

```
int A(){
    x =
(3) };
```

2. Based on the type definition for the node structure of a *linked list* to answer the following questions.

typedef struct NODE { int value; struct NODE *next;} LLNODE;

(i) Fill the blanks to complete the pseudo code for inserting an integer **X** as the last node of **aList**:

```
append ( int X, LLNODE &aList )
{   LLNODE   *newNode= new LLNODE;      // create a new node

    newNode ->value = X;                // store X into the new node
    newNode ->next = NULL;              // set the next pointer of the new node as NULL
    if (aList == NULL)                  // initially, aList is empty
        (4) aList -> next = NULL ;
    else                                // aList has one or more nodes
    {   LLNODE   *ptr = aList;          // the first node of aList

        while (ptr->next != NULL)
            ptr = ptr->next;            // locate the last node of aList
        (5) ptr -> next = NULL ;        // set the new node as the last node
    }   // end else
}   // end insert
```

(ii) Modify the type definition to transform it into a doubly linked list.

typedef struct NODE { int value; struct NODE *next; (6) struct NODE * pre **} LLNODE;**

(iii) For a doubly linked list, what statements should we add into **append**? Point out where to add them.

(7) 在 newNode -> next = NULL 的下面，ptr = ptr -> next 的下面