國立中正大學
110 學年度碩士班招生考試
試 題

[第 3 節]

科目名稱	軟體設計
系所組別	資訊工程學系-甲組

—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 110 學年度碩士班招生考試試題

1. (10%)

1.1 (4%) Will the following lines of C++ code compile?

a. (2%) int &* a1;　　Yes / No

b. (2%) int *& a2;　　Yes / No

1.2 (6%) Circle the outcome of the following lines of code with **const int* cip**.

a. (2%) const int** b1 = &cip;　　Compiles / Invalid syntax / Violates const-ness

b. (2%) int* const * b2 = &cip;　　Compiles / Invalid syntax / Violates const-ness

c. (2%) int ** const b3 = &cip;　　Compiles / Invalid syntax / Violate const-ness

2. (15%) Consider the following C++ classes. The code below causes no compiler errors.

```
#include <iostream>
using namespace std;

class A {
  public:
    void f1() { f2(); cout << "A::f1, "; }
    virtual void f2() { cout << "A::f2, "; }
  private:
    int m_ = 333; };

class B : public A {
  public:
    void f1() { cout << "B::f1, "; }
    virtual void f3() { cout << "B::f3, "; }
  private:
    int n_ = 451; };

class C : public B {
  public:
    virtual void f1() { cout << "C::f1, "; }
    void f2() { f3(); cout << "C::f2, "; }};
```

a. (3%) List out **in order** all functions (including **constructors** and **destructors**) that are called during the execution of the code below. Make sure to use the full **Class::Function** names.

```
A* ap = new B;
delete ap;
```

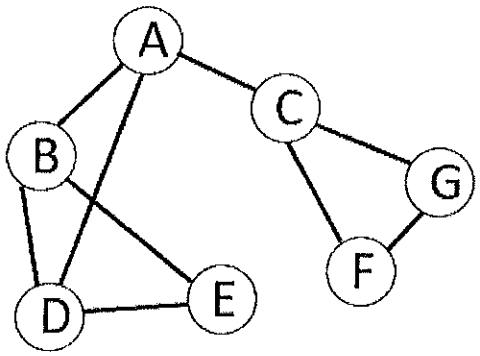
b. (12%) Assume we have objects and pointers as defined in the five lines of code below. Then, for each row of the table below, **fill in the result** on the right, which should either be the corresponding **stdout**

output, "compiler error", or "runtime error".

```
Line 1    B b;          // object instances
Line 2    C c;
Line 3    A *ap1 = &c;  // pointer
Line 4    B *bp1 = &b;
Line 5    B *bp2 = &c;
```

`bp1->f1();`	(甲)
`bp1->f2();`	(乙)
`bp2->f2();`	(丙)
`bp2->f3();`	(丁)
`ap1->f1();`	(戊)
`ap1->f3();`	(己)

3. (9%) To find the articulation points in the graph below, start with constructing a depth-first spanning tree by doing a depth-first search of the graph. Instead of choosing an arbitrary vertex to start, your first vertex must be set to vertex A, and select the smallest vertex in alphabetical order first when performing the depth-first search.



3.1 (2%) (　　) What are the edges of the depth-first spanning tree?

(a) (A, B), (B, D), (D, E), (A, C), (C, F), (C, G)
(b) (A, B), (A, D), (D, E), (A, C), (C, F), (F, G)
(c) (A, B), (B, D), (B, E), (A, C), (C, F), (C, G)
(d) (A, B), (A, C), (A, D), (D, E), (C, F), (C, G)
(e) (A, B), (B, D), (D, E), (A, C), (C, F), (F, G)

(f) None of the above

3.2 (2%) (　　) What are the depth-first numbers (*dfn*) of each vertex (in the order of A, B, C, D, E, F, G), when performing the depth-first search on the graph above?

(a) (1, 2, 3, 4, 5, 6, 7)
(b) (1, 5, 2, 6, 7, 3, 4)
(c) (1, 2, 5, 3, 4, 6, 7)
(d) (1, 4, 5, 2, 3, 7, 6)
(e) (1, 3, 2, 7, 6, 5, 4)
(f) None of the above

3.3 (3%) (　　) What are the output value of the low function (i.e., *low(v)*) for each vertex (in the order of A, B, C, D, E, F, G) based on (3.1) and (3.2)?

(a) (1, 2, 1, 2, 2, 5, 5)
(b) (1, 1, 5, 1, 2, 5, 5)
(c) (1, 1, 5, 1, 1, 5, 5)
(d) (1, 1, 5, 1, 4, 6, 7)
(e) (1, 2, 5, 2, 2, 5, 5)
(f) None of the above

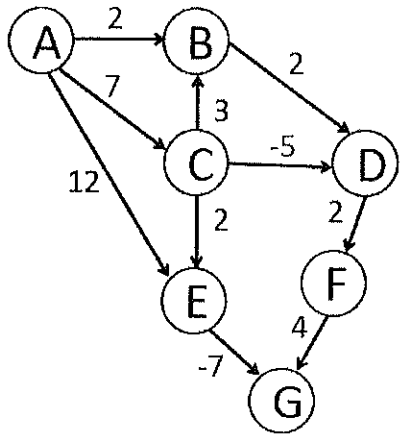
3.4 (2%) (　　) What are the articulation points in the graph?

(a) (A, C)
(b) (B, C)
(c) (A)
(d) (A, B, C)
(e) (A, E, G)
(f) None of the above

4. (3%) (　　) Given the in-order traversal (A, B, G, E, D, I, J, F, H, C) and the post-order traversal (G, E, J, I, H, F, D, C, B, A), find the binary tree. What is the pre-order traversal sequence of this binary tree?

(a) (B, A, C, D, E, G, F, H, J, I)
(b) (A, B, C, H, G, E, D, I, F, J)
(c) (A, B, C, D, E, G, F, I, J, H)
(d) (A, C, F, I, J, G, E, H, D, B)
(e) (A, B, G, C, E, H, D, I, F, J)
(f) None of the above

5. (8%) Given a weighted directed graph with negative edges, please answer the following questions.



5.1 (3%) (　　) Please perform Dijkstra's algorithm running on the graph above from vertex A. What are the vertices in the order they are selected by Dijkstra's algorithm?

(a) (A, B, D, F, C, G, E)
(b) (A, B, D, F, C, E, G)
(c) (A, B, C, E, D, F, G)
(d) (A, B, D, F, G, C, E)
(e) (A, B, C, E, G, D, F)
(f) None of the above

5.2 (3%) (　　) What is the cost of the shortest path from vertex A to each vertex (in the order of A, B, C, D, E, F, G), after performing Dijkstra's algorithm?

(a) (0, 2, 7, 2, 9, 6, 2)
(b) (0, 2, 7, 2, 9, 4, 2)
(c) (0, 2, 7, 2, 9, 4, 8)
(d) (0, 2, 7, 2, 12, 6, 10)
(e) (0, 2, 7, 4, 9, 6, 2)
(f) None of the above

5.3 (2%) (　　) What is the resulting shortest path from vertex A to vertex D after performing Dijkstra's algorithm?

(a) A->B->D
(b) A->C->D
(c) A->E->G->D
(d) A->C->E->G->D
(e) None of the above

6. (5%) (　　) Consider the following failure function and a string "acacabacacabacacac" with 18 characters. What are the values of *f(0), f(3), f(5), f(10), f(13), f(17)*?

$$f(j) = \begin{cases} largest\ k < j\ such\ that\ p_0 p_1 \ldots p_k = p_{j-k} p_{j-k+1} \ldots p_j\ if\ such\ a\ k\ \geq 0\ exists \\ -1, \quad \text{otherwise} \end{cases}$$

(a) (-1, 1, -1, 4, 7, 3)
(b) (-1, -1, 0, 1, 2, 3)
(c) (-1, 1, 3, 5, 8, -1)
(d) (-1, -1, -1, 4, 7, -1)
(e) (-1. 2, 4, 0, 3, 7)
(f) None of the above

7. (4%) Use substitution method to find a tight asymptotic upper bound for this recurrence:
T(n)=T(n/4)+T(n/5)+6n

8. (4%) Use substitution method to find a tight asymptotic lower bound for this recurrence:
$T(n)=4T(n/2)+n^2$

9. (3%) Prove or disprove with a counter example that every comparison based sorting algorithm has time complexity Ω(nlgn).

10. (3%) If a dynamic programming problem satisfies the property of optimal substructure, justify if a locally optimal solution is also globally optimal. A simple yes/no answer will not get any point.

11. (5%) Let G = (V, E) be an acyclic directed graph with negative-weight edges, argue if Johnson's reweighting approach can solve a single source shortest path problem in G faster than the Bellman-Ford algorithm. A simple yes/no answer will not get any point.

12. (6%) In the knapsack problem, given k items whose weight w_i and the corresponding value v_i has the relationship: $w_1>w_2>w_3>\ldots>w_k$ and $v_1<v_2<v_3<\ldots<v_k$. If we want to develop an efficient algorithm to find the optimal solution for such problem, determine if the two types of knapsack, (a) fractional knapsack, and (b) 0-1 knapsack, have the following properties: (1) optimal substructure; (2) greedy choice; (3) overlapping subproblems

13. (10%) Please correct errors in the following program. You need to explain your corrections.

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

```
void main()
{
        int X;
        scanf("%d", X);

        if ( X&2 == 0) {
                printf("%d is an even number\n", X);
        }
        else {
                printf("%d is an odd number\n", X);
        }
}
```

14. (10%) Please write a function to find a key in a BST (Binary Search Tree). Return the address of the matched node if there is a match. Otherwise, return NULL;
    The data structure for a tree node contains a key and two pointers. The key is a string.
    You need to declare the data structure of the tree node before the function code.

15. (5%) Please describe the following two string functions: You need to describe its functionality as well as the data type of the parameters and the return value.
    a) (2%) strcmp()
    b) (3%) strdup()