國立中正大學
108 學年度碩士班招生考試
試 題

[第 2 節]

系所組別	資訊工程學系－甲組
科目名稱	軟體設計

－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。

1. (25%) Questions 1.1 to 1.11. Choose the best answer. You can generally assume that code shown in the questions is intended to be syntactically correct, unless something in the question or one of the answers suggests otherwise.

1.1 (2%) (　) cout is what?
(a) an object (a variable whose type is a class) (b) a library function (c) a system file
(d) a C++ operator (e) a class

1.2 (2%) (　) Which represents a proper use of the new C++ type for Boolean values?
(a) char errorFlag = "true"; (b) bool errorFlag = true; (c) bool errorFlag = FALSE;
(d) enum errorFlag = {true, false}; (e) Boolean errorFlag;

1.3 (2%) (　) A **private** member of the class is accessible in
(a) only in the current class (b) same class and derived class (c) outside the class
(d) none of the above

1.4 (2%) (　) Which of the following functions will increment a double passed as argument to the functions?

```
void inc1 (double &a) { a++; }
void inc2 (double *p) { p++; }
void inc3 (double *p) { (*p)++; }
void inc4 (const double &a) { a++; }
```

(a) function inc1 and inc2 (b) functions inc1 and inc3 (c) function inc1 and inc4
(d) function inc3 and inc4 (e) none of the above

1.5 (2%) (　) Choose the respective **delete** operator usage for the expression "ptr = new int [100]".
(a) delete ptr; (b) delete ptr [100]; (c) delete [ ] ptr; (d) [100] delete ptr;

1.6 (2%) (　) Which of the following function declaration is/are correct?

```
int sum1 (int a=1, int b=2, int c=3);
int sum2 (int a=1, int b);
int sum3 (int a=1, int b, int c=3);
```

(a) sum1, sum2, and sum3 (b) sum1 and sum2 (c) sum1 and sum3
(d) all are correct (e) none is correct

1.7 (2%) (　) The copy constructor is executed on:
(a) assigned one object to another object at its creation
(b) when objects are sent to functions using call by value mechanism
(c) when the function returns an object

(d) all of the above (e) none of the above

1.8 (2%) ( ) Choose the pure virtual function definition form the following:
(a) virtual void f() = 0 { }; (b) void virtual f() = 0 { }; (c) virtual void f() { } = 0;
(d) none of the above

1.9 (3%) ( ) int i;
double d;
char c;
char str[100];
cin >> i >> d >> c >> str;
cout << str;
Suppose at the point where the statement with cin above is executed, the user types in
30 40 King Dome
(The leading spaces are part of what is typed in.) What will be the result displayed by the line with cout?
(a) cannot tell, because an error would have occurred reading in the variable i
(b) King Dome (c) King (d) ing Dome (e) ing

1.10 (3%) ( ) Trace through the operation of this code, and then pick the true statement:

```
#include <iostream>
void f (int& p1, int& p2) { p1 = p2 + 7 * 2; }
int main (void) {
int x, y;
x =2 ; f (x, x);
y = x ; f (y, y);
cout << x << endl; return 0; }
```

(a) it prints the value 18 (b) y's value after the call f (y, y) is 16
(c) it prints the value 30 (d) it prints the value 16 (e) it prints the value 2

1.11 (3%) ( ) The filestream decalred as

```
ifstream iFile;
```

has been opened successfully. This line of the program

```
iFile >> count;
```

executes, where count is an integer variable. However, the file at this point contains the letter 'x' rather than a valid integer. Which code properly detects such an error?
(a) if (iFile) { // OK
} else { //error
}

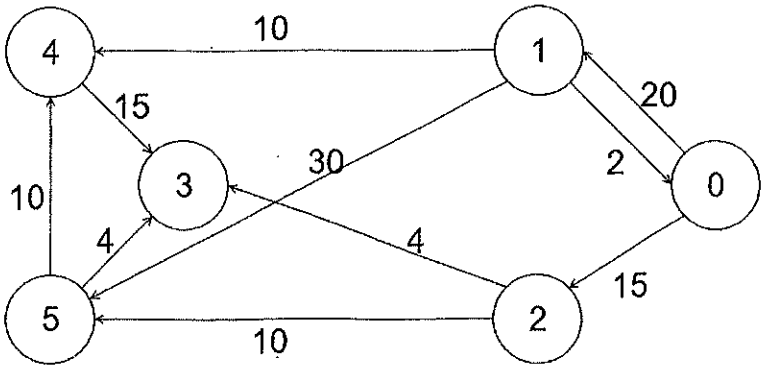
(b) if (iFile[0] != int) { // error…
}
(c) char chCount = char (count);
if (isalpha(chCount)) { // error…
}
(d) if (isalpha(count)) { // error…
}
(e) if (ifstream.error()) { // error…
}

2. (8%) To build a binary search tree, the following elements are inserted in the given order: 65, 50, 80, 70, 60, 62, 90, 10. You have to use the following struct *treeNode* with one additional filed *leftSize*, where the field *leftSize* of a tree node *u* denotes the number of tree nodes in the *u*'s left subtree plus one. For example, if the number of nodes in the left subtree of the tree node *u* is 3, then the tree node *u*'s field *leftSize* is equal to 4.
   (a) (3%) Please draw the binary search tree, and each tree node stores its element and *leftSize*.
   (b) (5%) Please illustrate how to obtain the tree node storing the *k*-th smallest element in the binary search tree built in (a) from the tree root, and explain your procedure. Note that you have to use the additional field *leftSize* and the time complexity of your procedure should be $O(\log n)$, where n is the number of input elements.

```
typedef struct _tree_node_ *treePointer;
typedef struct _tree_node_ {
    treePointer leftChild;
    treePointer rightChild;
    int element;
    int leftSize;
} treeNode;
```

3. (6%) To build a max heap, the following key values are inserted in the given order: 7, 16, 49, 82, 5, 31, 6, 2, 44.
   (a) (3%) Please write down the status of the max heap after **all** values are inserted into the max heap.
   (b) (3%) Starting with the max heap built in (a), please write down the status of the max heap as well as the sorted array after the ***5-th*** iteration during the heap sort. Note that each iteration swaps the first and the last records in the current heap, and then rearranges records so that the record with maximum key is moved to its correct position in the sorted array and the remaining heap is still a max heap.

4. (4%) Given an edge-weighted directed graph, please answer the following questions.



(a) (2%) Perform Dijkstra's algorithm to determine the shortest path from vertex 0 to each of the other vertices. Please list the vertices in the order they are processed by the algorithm.

(b) (2%) Perform a breadth-first traversal of the graph, starting with vertex 0. Select the smallest edge first, when appropriate.

5. (2%) Assume that we have a 17-bucket hash table with one slot per bucket. The hash function for each key using the division hash function, *key*%17. Sequentially insert the records with keys are 6, 12, 34, 29, 28, 11, 23, 7, 0, and 33. Please complete the hash table at the end with Linear Probing to handle overflows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

6. (5%) Please answer the following questions.

(a) (2%) What is the maximum number of tree nodes in a k-ary tree of height h?

(b) (3%) Prove your answer in (a).

7. (6%, True or false. If the statement is false, correct the wrong part. Simply negate the statement is not accepted.)

(a) Finding a longest common subsequence from an arbitrary number of input sequences is a P problem.

(b) Given a non-negatively weighted graph G and two vertices u and v, finding the longest simple path from u to v is a NP-hard problem.

(c) Breadth-first search can explore all the edges in a graph G and classify them.

8. (5%, 1 extra point deduction for an incorrect answer until 5 points) Which of the following sorting algorithms are stable in nature? (a) Counting sort; (b) Quicksort; (c) Mergesort; (d) Insertion sort; (e) Heapsort.

9. (8%) Determine if the following sorting algorithms are asymptotically optimal or not: (a) Bubble sort; (b) Quicksort; (c) Mergesort; (d) Heapsort.

10. (1%) What is the name of such representation when a directed graph with edges (u→v) is defined as follows?

v \ u	1	2	3
1	0	0	0
2	20	0	-30
3	40	0	0

11. (1%, 4%) Following the above question, what does Johnson's algorithm do to such graph, and what is the result? Write down your process.

12. (10%) What are the outputs of the following codes.

a) (3%)
```
#include <stdio.h>
void main() {
  int x = 2;   int y = 10;
  if (x > y)
      if (y > 0)
        printf("B");
  else
      printf("H");

  printf(" or C");
  return 0; }
```

b) (7%)
```
#include <stdio.h>
void main() {
  struct node {
    char *name;
    double x;
    long long length;
  };

  char text[100];   char *p;   char *q;   int i;

  memset(text, 0, 100);
  p = text;
  for (i=0; i<10; i++) { *p++ = 'a' + i; }
```

```
q = text + 3;
for (i=0; i<3; i++) { *p++ = *q++; }

printf("%d\n", sizeof(struct node));
printf("%d\n", sizeof(short));
printf("%s\n", q); }
```

13. (7 %) Please correct the errors in the following program such that the output of the following program is: AppleAppleApple

```
#include <stdio.h>
#include <string.h>
void main() {
    int *p;    char text[100];    char *q;
    strcpy(q, "Apple") ;    // copy "Apple" to q
// We want to append three "Apple" to text such that the result of text will be "AppleAppleApple"
    for (i=0, i<3, i++) {
      strcpy(text, q);
      text += strlen(q); }
    printf("%s\n", text); }
```

14. (8%) a) (2%) Please point out the error in the following declaration of array A.
b) (6%) What is the final content of A after the loop is executed?

```
int   A[] = (0, 1, 2, 3, 4, 5);   // declare an array of six integers with initial values
char *p;
p = A;
p = &A[*(p+2)];
for (i=0; i<3; i++) { *p = *p + i;   p++; }
```