國立臺北科技大學103學年度碩士班招生考試

系所組別：2300資訊工程系碩士班

第三節　程式設計　試題

注意事項：

1. 本試題共8題，配分共100分。
2. 請標明大題、子題編號作答，不必抄題。
3. 全部答案均須在答案卷之答案欄內作答，否則不予計分。

Problem 1 [8%, each 2%]

Please give the best asymptotic running time for each of the problems or bound for each of the recurrences shown below using the "big oh" notation. It is assumed that $T(1)=d$ for some constant d and c is a constant in all the recurrences. Just state the answer - you **do NOT** need to justify them.

(1) $T(n)=2T(n/2)+cn^2$

(2) $T(n)=\sqrt{n}T(\sqrt{n})+n$

(3) Finding the median in an unsorted set of size n.

(4) In a directed, weighted graph $G=(V, E)$ with positive weights and $|V|=n$ and $|E|=m$, determine the shortest path between a given pair of vertices.

Problem 2 [8%, each 2%]

Mark each of the following by **T** (=true) or **F** (=false):

(1) If any NP-complete problem is not in P, then no NP-complete problem is in P.

(2) Suppose problem P_1 can be reduced to problem P_2 in linear time. Then, if P_2 is NP-hard then P_1 is NP-hard.

(3) If algorithm A solves problem L in $O(n \log n)$ time, then no instance of problem L can, when given to A as input, make A take n^2 time.

(4) If problem P has an $\Omega(n^2)$ lower bound then there can not exist an algorithm A that solves P in $O(n \log n)$ time.

注意：背面尚有試題

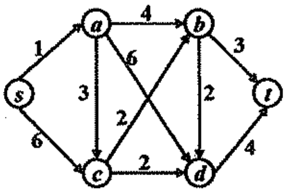
Problem 3 [12%, each 4%]

Consider the edge-weighted connected graph $G = (V, E)$ in the following figure where V is the vertex set and E is the edge set of G, respectively.

(1) Please find the shortest path from vertex s to t by using the Dijkstra's algorithm. Show your work step by step.

(2) Does the Dijkstra's algorithm allow the edges having negative weight in the input graph? Please give the reasons to support your conclusion.

(3) Give an algorithm that determines whether or not a given undirected graph G=(V, E) contains a cycle. Your algorithm should run in O(|V|) time, independent of |E|.



Problem 4 [21%, each 3%]

Given the program below in C. Please trace the program and fill the 4-1~4-7 blanks with the printf output of each statement.

```
#include <stdio.h>

void test01(int a, int b, int c) {
   float result;
   result = (a+b/c-a%b)/4.0;
   printf(″%.1f\n″, result);                    /*   Problem 4-1    */
}

void test02(int a, int b){
   int shift;
   if (!(0<= a && b<=100))
      shift = a>>1 | b<<1;
   else
      shift = a<<1 | b>>1;
   printf(″%d\n″, shift);                       /*   Problem 4-2    */
}

void test03(int a[], int size){
    int x=0, i, j;
    for (i=0; i<size; i++){
        for (j=0; j<i; j++)
            x+=a[j];
    }
    printf(″%d\n″, x);                          /*   Problem 4-3    */
}

void test04(int a[], int size) {
    enum GRADE {FRESHMAN=1, SOPHOMORE, JUNIOR, SENIOR};
    int sum=0, i, level;
    for(i=0; i<size; i++) {
        switch (a[i]){
            case FRESHMAN: level=1;
            case SOPHOMORE: level=2;
                            break;
            case SENIOR: level=3;
            case JUNIOR: level=4;
                         break;
            default: level=0;
                     break;
        }
        sum = sum+ level;
    }
    printf(″%d\n″, sum);                        /*   Problem 4-4    */
}
```

```
void test05(int *p, int *q, int *r) {
    int **pp=&p, **qq=&q, **rr=&r;
    qq = &p;  *rr = q;
    printf(″%d\n″, **qq + **rr);                /*   Problem 4-5   */
}

void test06(char *list[], int size) {
    int i=0, first=0;
    while(i<size){
        if(strcmp(list[i], list[first]) < 0)
           first = i;
        i++;
    }
    printf(″%s\n″, list[first]);               /*   Problem 4-6    */
}

int test07(int n) {
    int ans;
    if (n==1)
       ans=0;
    else
       ans = 1 + test07(n/2);
    return (ans);
}
int main(int argc, char *argv[]) {
    int a=9, b=5, c=3, array[]={1,2,3,4,5};
    char *s[]={″happy″, ″birthday″, ″new year″, ″memorial day″, ″festival″};
    test01(a, b, c);
    test02(a, b);
    test03(array, 3);
    test04(array, 5);
    test05(&array[0], &array[1], &array[2]);
    test06(s, 5);
    printf(″%d\n″, test07(5));                    /*   Problem 4-7   */
    return 0;
}
```

Problem	Answer
4-1	
4-2	
4-3	
4-4	
4-5	
4-6	
4-7	

Please copy the above answer table to your answer sheet.

Problem 5 [15%, each 3%]

Please trace the following C++ program and provide the results of the cout statements.

```
#include <iostream>
using namespace std;
class Animal {
    public:
    virtual void speak()=0;
};
class Bird : public Animal {
    public:
    virtual void speak(){  cout << "twitter" <<endl;    }
};
class Mammal : public Animal {
    public:
    virtual void speak( ) { cout << "can't speak" <<endl; }
    void bark( ) { cout << "can't bark" <<endl; }
};
class Cat : public Mammal {
    public:
    void speak( ) { cout << "meow!"; }
    virtual void purr( ) { cout << "purrrrr" <<endl; }
};
class Dog : public Mammal {
public:
    virtual void speak( ) { cout << "woof!" <<endl; }
    void bark( ) { cout << "wooooof!" <<endl; }
};
int main(int argc, char *argv[]) {
    Dog * d = new Dog( );
    Mammal * m = d;
    m->bark( );                                    // problem 5-1
    Animal * a = m;
    a->speak( )                                    // problem 5-2
    Mammal mm = *d;
    mm.speak( );                                   // problem 5-3
    mm.bark();                                     // problem 5-4
    Animal * aa = new Cat( );
    Bird *b = dynamic_cast<Bird *> (aa);
    char cast_result[25];
    if( b )       strcpy(cast_result, "Variable was a Bird");
    else          strcpy(cast_result, "It was not a Bird");
    cout << cast_result << endl;                   // problem 5-5
    return 0;
}
```

Problem	Answer
5-1	
5-2	
5-3	
5-4	
5-5	

Please copy the above answer table to your answer sheet.

注意：背面尚有試題

Problem 6 [12%, each 3%]

The picture below shows the output result of the following main() program in which the printUp(int) function prints the upper half of the picture and the printDown(int) function prints the lower half. Please answer problems 6-1~6-4 with the correct statements.

Input	Output	Input	Output
3	[redacted]	5	[redacted]
10	Error! (1, 3, ..,~9)		

```
#include <stdio.h>
#include <stdlib.h>
void printUp(int num){
    int i=0,k=0;
    for(k = 0 ; k < num ; k++){
        for(i=1;i< num-k;i++)
            printf(".");
        for(i=0; i<=k; i++)
            printf("%d",_______);    //(6-1)
        for(i=0; i<k; i++)
            printf("%d", num-k+i+1);
        for(i=1; i< num-k; i++)
            printf(".");
        printf("\n");
}
void printDown(int num) {
    int i=0, k=0;
    for(k=0;k<_______;k++){          //(6-2)
        for(i = 0 ; i <= k; i++)
            printf(".");
        for(i=0; i<num-k-1; i++)
            printf("%d",_______);    //(6-3)
        for(i = 0 ; i < num-k-2; i++)
            printf("%d",_______);    //(6-4)
        for(i = 0 ; i <= k; i++)
            printf(".");
        printf("\n");
    }
}
```

```
int main() {
    int c=0, num=0;
    scanf("%d", &num);
    if(num<1||num>9||num%2==0){
      printf("Error! (1,3,5,7,9)\n");
    }
    else {
        printUp(num);
        printDown(num);
    }
    return 0;
}
```

Problem	Answer
6-1	
6-2	
6-3	
6-4	

Please copy the above answer table to your answer sheet.

Problem 7 [12%, each 3%]

In mathematical numeral systems, the base is the number of unique digits to represent numbers. For example, the base is 10 for the decimal system because it uses the ten digits from 0 through 9. Consider the following C program that can transform a numeral system into another. The inputs and outputs of this program are given below.

Input	Output	Note
1111 2 10	1111_(2) =15_(10)	$(1111)_2 = (15)_{10}$
15 10 2	15_(10) =1111_(2)	$(15)_{10} = (1111)_2$
234 5	234_(5) =1000101_(2)	$(234)_5 = (1000101)2$

Please trace this C program and answer problems 7-1~7-4 with the correct statements.

```
#include <stdio.h>
#include <math.h>
// Transfer x_(base1=10) to result(base2)
// Transfer x_(base1) to result(base2=10)
int transform10 (int x, int base1, int base2){
   int remainder=0;
   int result=0;
   int index=1;
   while (x>________) {                             // (7-1)
     remainder = x % __________;                    // (7-2)
     result = result + remainder * __________;      // (7-3)
     index = index * __________;                    // (7-4)
     x = x / base2;
   }
   return result;
}
int transform(int x, int base1, int base2) {
    int temp=transform10(x, base1, 10);
    return transform10(temp, 10, base2);
}
int main(){
    int x;
    int base1=0, base2=0;
    scanf("%d %d %d",&x, &base1, &base2);
    printf("%d_(%d) =%d_(%d)\n",x, base1, transform(x,base1,base2), base2);
    return 0;
}
```

Problem	Answer
7-1	
7-2	
7-3	
7-4	

Please copy the above answer table to your answer sheet.

Problem 8 [12%, each 2%]

Consider the following C++ program that implements a binary tree structure. Please trace this C++ program and answer problems 8-1~8-6 with the correct statements. The output of this program is: 5 7 9 10 16 28

```
#include <iostream>
using namespace std;
class Node {
    private:
    int _data;
    public:
    Node* left, * right;
    Node(int data) {
        _data = data;
        left = right = NULL;
    }
    ~Node() {       }
    int getData() { return _data;  }
};
class Tree {
    Node* root;
    void destroy(Node *leaf);
    void insert(int key, Node *leaf);
    public:
    Tree();
    void insert(int key);
    void destroy(){ destroy(root);}
    void print();
};
Tree::Tree() {   root=NULL; }
Tree::~Tree() { destroy(); }
void Tree::destroy(Node *leaf) {
  if(leaf!=NULL)    {
    destroy(leaf->left);
    destroy(leaf->right);
    delete leaf;
  }
}
void Tree::insert(int key, Node *p) {
  if(key< ____________)    {       // (8-1a)
    if(p->left!=NULL)
     insert(key, p->left);
    else      {
      p->left=new __________;    // (8-2a)
      p->left->left=NULL;
      p->left->right=NULL;
    }
  }
```

```
else if(key >= ____________) {    //(8-1b)
    if(p->right!=NULL)
      insert(key, p->right);
    else      {
      p->right=new __________;    //(8-2b)
      p->right->left=NULL;
      p->right->right=NULL;
    }
  }
}
void Tree::insert(int key) {
  if(root!=NULL)
    insert(__________);            //(8-3)
  else    {
    root=new Node(key);
    root->left=root->right=_______; //(8-4)
  }
}
void Tree::print() {
     Node *p = root;
     Node * stack[20];
     int top=-1;
     while (1) {
         if (p==NULL) {
             if (top<0) break;
             p = stack[________];   //(8-5)
             cout<<p->getData()<<" ";
             p = p->right;
         }
         else {
             stack[_________]=p;    //(8-6)
             p = p->left;
         }
     }
}
int main() {
    int data[] = {5, 16, 7, 28, 9, 10};
    Tree t;
    for(int i=0;i<6;i++) t.insert(data[i]);
    t.print();
    t.destroy();
    return 0;
}
```

Problem	Answer
8-1a、8-1b	
8-2a、8-2b	
8-3	
8-4	
8-5	
8-6	

Please copy the above answer table to your answer sheet.