

國立臺北科技大學 101 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 軟體設計 試題

第一頁 共四頁

注意事項：

1. 本試題共 5 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 [27%, each 3%]

Given the program below in C, please trace the program and fill blanks with the printf output of each statement.

```
#include<stdio.h>
#define SQUARE(n) n * n

int test01(int v){
    int ans;

    if (v < 1) ans = 0;
    else if (v % 3 == 0) ans = 0 + test01(v/3);
    else if (v % 3 == 1) ans = 1 + test01(v/3);
    else if (v % 3 == 2) ans = 2 + test01(v/3);

    return (ans);
}

int test02(){
    typedef enum {red, orange, yellow, green, blue, indigo, violet} color_t;
    typedef enum {sunday, monday, tuesday, wednesday, thursday, friday, saturday} day_t;

    int sum = 0;
    switch (indigo) {
        case monday:    sum += monday;
        case tuesday:   sum += tuesday;
        case wednesday: sum += wednesday;
        case thursday:  sum += thursday;
        case friday:    sum += friday;
        default:        sum += saturday;
    }

    return sum;
}
```

```
int test03(){
    int i = 0, j;
    do {
        for (j=0; j<4; j++)
            i = i + 1;
    } while (i < 5);
    return i;
}
```

```
int test04(){
    return strcmp("49", "5");
}
```

```
int test05(){
    int a[] = {1, 2, 3, 4, 5};
    int *p = &a[1];
    p++;
    return *p;
}
```

```
int test06(int v){
    v &= 1;
    v |= 0;
    v <<= 3;
    v >>= 1;
    return v;
}
```

```
int test07(int a, int b){
    return SQUARE(a + b);
}
```

```
int test08(){
    return 1 || 0 && 0;
}
```

```
int test09(int a, int b){
    int ans = 0;

    if (a < 0)
        if (b < 6)
            ans = 0;
    else
        ans = 1;

    return ans;
}
```

```
int main()
{
    printf("%d\n", test01(15)); /* Problem 1-1 */
    printf("%d\n", test02()); /* Problem 1-2 */
    printf("%d\n", test03()); /* Problem 1-3 */
    printf("%d\n", test04()); /* Problem 1-4 */
    printf("%d\n", test05()); /* Problem 1-5 */
    printf("%d\n", test06(15)); /* Problem 1-6 */
    printf("%d\n", test07(9, 5)); /* Problem 1-7 */
    printf("%d\n", test08()); /* Problem 1-8 */
    printf("%d\n", test09(1, 2)); /* Problem 1-9 */

    system("pause");
    return 0;
}
```

Problem	Answer
1-1	
1-2	
1-3	
1-4	
1-5	
1-6	
1-7	
1-8	
1-9	

Please copy the above answer table to your answer sheet.

注意：背面尚有試題

Problem 2 [12%, each 2%]

C uses pointers for access to dynamically allocated memory such as the nodes of a linked list. Sometimes a pointer variable must be passed as an argument to a function. Consider the following C program that implements the stack data structure using a linked list. Please trace this C program and fill the blanks with correct statements.

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

typedef int stack_element_t;
typedef struct stack_node_s {
    stack_element_t    element;
    struct stack_node_s *restp;
} stack_node_t;

/* The value in ele is placed on top of the stack implemented as
 * a linked list of nodes accessed through top
 * input/output formal parameter: top
 * input formal parameter: value (element to add) */
void push(_____top, stack_element_t value) { /* Problem 2-1, data type of top */
    stack_node_t *newp; /* pointer to new stack node */

    /* Creates and defines new node */
    newp = (stack_node_t *) malloc(sizeof(stack_node_t));
    newp->element = ele;
    newp->restp = _____; /* Problem 2-2 */

    /* Sets stack top pointer to point to new node */
    _____ = newp; /* Problem 2-3 */
}

/* Print stack element from the top of the stack */
void print_stack(_____p) /* input - stack */ /* Problem 2-4, data type of p */
{
    if (p == NULL) {
        printf("Reach stack bottom.\n");
        return;
    }
    else {
        printf("%d\n", _____); /* print the node element pointed by p */ /* Problem 2-5 */
        p = _____; /* let p point to next node in the stack */ /* Problem 2-6 */
        print_stack(p);
    }
}

int main(void) {
    stack_node_t *stackp = NULL; /* stack of integers - initially empty */
    push(&stackp, 101); /* build the stack */
    push(&stackp, 202);
    print_stack(stackp);
    return 0;
}
```

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

Please copy the above answer table to your answer sheet.

Problem 3 [18%, each 3%]

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

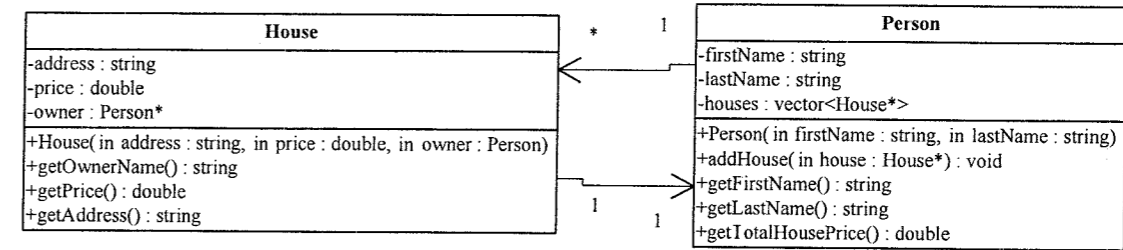
```

#include <iostream>
#include <vector>
using namespace std;
class Observer;
class Subject {
public:
    ~Subject();
    Subject() {cost = 10; profit = 20; }
    void addObserver(Observer* ob);
    void notify();
    int inc() { return (++id); }
    int cost;
    int profit;
private:
    vector<Observer*> obs;
    static int id;
};
int Subject::id=0;
class Observer{
protected:
    Subject* subject;
    int id;
public:
    virtual void view() = 0;
    void attach(Subject* b) {
        subject = b;
        id = b->inc();
        subject->addObserver(this);
    }
};
Subject::~Subject() {
    int size = obs.size();
    if (size > 0){
        obs.erase(obs.begin(),
            obs.begin() + size );
    }
}

void Subject::addObserver(Observer* ob){
    obs.push_back(ob); //Add ob. at the end
}
void Subject::notify(){
    for (vector<Observer*>::iterator
        it = obs.begin();
        it != obs.end(); it++){
        Observer* ob = *it;
        ob->view();
    }
}
class Boss:public Observer {
public:
    void view();
};
void Boss::view() {
    cout << id << ":Boss\n";
    cout << subject->profit << "\n";
}
class Employee:public Observer {
public:
    void view();
};
void Employee::view() {
    cout << id << ":Employee\n";
    cout << subject->cost << "\n";
}
int main() {
    Boss *a1 = new Boss;
    Observer *a2 = new Boss;
    Employee *a3 = new Employee;
    Subject *b = new Subject;
    a1->attach(b);    a2->attach(b);
    a3->attach(b);    b->notify();
    delete a1; delete a2;
    delete a3; delete b; return 0;
}
    
```

Problem 4 [22%]

A house management application with two classes House and Person is designed based on the following class diagram.



(a) Please finish the declaration (.h only) of the following Person class in C++. [7%]

```

class Person {
public:
    ...
};
    
```

(b) A person may have several houses. Each house has a price. The getTotalHousePrice() function computes the total price of the houses that a person owns. Please give a C++ implementation of the getTotalHousePrice() function (.cpp only). [7%]

(c) By using House and Person classes, we would like to create 3 houses and 2 persons with the following relationships. The person "Mary Chen" has two houses. The first house is located at "address1" and worth 1,000,000 dollars, and the second house is located at "address2" and worth 1,200,000 dollars. The person "David Hwang" has a house, which is located at "address3" and worth 2,000,000 dollars. Please give the C++ client code that can create the instances of such 3 houses, 2 persons, and the associations between the houses and the persons. [8%]

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

Please copy the above answer table to your answer sheet.

注意：背面尚有試題

