

## 國立臺北科技大學 107 學年度碩士班招生考試

系所組別：2300 資訊工程系碩士班

## 第二節 程式設計 試題

第一頁 共三頁

**注意事項：**

1. 本試題共五題，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 [12%, each 3%]**

Given below program in C. What are the correct answers for the following 1-1~1-4 questions?  
Provide your answers in the answer table and copy the table to your answer sheet.

| Line No |                          |
|---------|--------------------------|
| 1       | int test(int a, int b) { |
| 2       | int result = 0;          |
| 3       | if ((a<3) && (b<7)) {    |
| 4       | if (b==4) {              |
| 5       | result = (a-b);          |
| 6       | }                        |
| 7       | else {                   |
| 8       | result = a*b;            |
| 9       | }                        |
| 10      | }                        |
| 11      | else {                   |
| 12      | result = (a+b);          |
| 13      | }                        |
| 14      | return result;           |
| 15      | }                        |

| Problem | Answer |
|---------|--------|
| 1-1     |        |
| 1-2     |        |
| 1-3     |        |
| 1-4     |        |

Please copy the above answer table to your answer sheet.

- 1-1. Which of the following sets of input data can execute all of the program statements at least once?  
(A) {test(4, 4), test(4, 5), test(2, 5)} (B) {test(2, 4), test(4, 5), test(2, 5)}  
(C) {test(4, 4), test(1, 5), test(2, 5)} (D) {test(2, 4), test(1, 5), test(2, 5)}
- 1-2. Which of the following sets of input data can execute both **true** and **false** branches for all of the "if" statements at least once?  
(A) {test(4, 4), test(2, 4), test(2, 5)} (B) {test(2, 4), test(5, 2), test(3, 7)}  
(C) {test(4, 4), test(1, 5), test(2, 5)} (D) {test(2, 4), test(1, 5), test(2, 5)}

- 1-3. Which of the following program statements (in terms of line number) will NOT be executed by the set of input data {test(2, 4), test(1, 5), test(2, 5)}?  
(A) 5, 8 (B) 8, 12 (C) 12 (D) 5, 8, 12 (E) 5

- 1-4. If the program statement at Line 12 is modified to "result = a + b + test(a/2, b/2);", what value will be returned by test(8, 8)? (A) 4 (B) 8 (C) 16 (D) 28 (E) 30

**Problem 2 [30%, each 3%]**

Given below program in C. Please trace the program and provide the output of each printf statement in problems 2-1~2-10.

|   |  |
|---|--|
| <pre>#include&lt;stdio.h&gt; #include&lt;string.h&gt; float test01(void){     int a = 5, b = 2;     return a / b; } int test02(void){     enum {Spring, Summer, Fall = 5, Winter}     season;     season = Spring + Summer + Fall + Winter;     return season; } int test03(void){     int n = 1;     for (;){         n *= 5;         if (n &gt; 55) break;     }     return n; } int test04(void){     struct {char str1[10], str2[10];} s;     strcpy(s.str1, "yahoo");     strcpy(s.str2, "google");     strcat(s.str1, strcat(s.str2, "adobe"));     return strcmp(s.str1, s.str2); } int test05(void){     int a[] = {10, 20, 30, 40, 50};     int *p = &amp;a[sizeof(a)/sizeof(int) - 1], *q = a;     return (p - q); } int test06(void){     int digits = 0, n = 12345;</pre> | <pre>do {     n /= 10;     digits++; } while (n &gt; 0); return digits; } int test07(void){     int grade = 3, ans;     switch (grade) {         case 4: ans = 4;         case 3: ans = 3;         case 2: ans = 2;         case 1: ans = 1;         default: ans = 0;     }     return ans; } int test08(void){     int n = 0, sum = 0;     while (n &lt; 5) {         n++;         if (n &lt; 3) continue;         sum += n;     }     return sum; } unsigned char test09(void){     unsigned char a[] = {0xFF, 0x00, 0x01,                         0x08};     a[0] &amp;= 1;    a[1]  = 1;     a[2] &lt;&lt;= 3;   a[2] &gt;&gt;= 3;     return (a[0] + a[1] + a[2] + a[3]); } int test10(void){     int a[][3] = {{6,7,8}, {9,10,11}, {12,13,14}};     return *&amp;a[1][1]; }</pre> |
|---|--|

注意：背面尚有試題

```
int main(){
    printf("%3.1f\n", test01()); /* Problem 2-1 */
    printf("%d\n", test02()); /* Problem 2-2 */
    printf("%d\n", test03()); /* Problem 2-3 */
    printf("%d\n", test04()); /* Problem 2-4 */
    printf("%d\n", test05()); /* Problem 2-5 */
    printf("%d\n", test06()); /* Problem 2-6 */
    printf("%d\n", test07()); /* Problem 2-7 */
    printf("%d\n", test08()); /* Problem 2-8 */
    printf("%u\n", test09()); /* Problem 2-9 */
    printf("%d\n", test10()); /* Problem 2-10 */
    return 0;
}
```

| Problem | Answer |
|---------|--------|
| 2-1     |        |
| 2-2     |        |
| 2-3     |        |
| 2-4     |        |
| 2-5     |        |
| 2-6     |        |
| 2-7     |        |
| 2-8     |        |
| 2-9     |        |
| 2-10    |        |

Please copy the above answer table to your answer sheet.

**Problem 3 [30%, each 3%]**

| Problem | Answer |
|---------|--------|
| 3-1     |        |
| 3-2     |        |
| 3-3     |        |
| 3-4     |        |
| 3-5     |        |
| 3-6     |        |
| 3-7     |        |
| 3-8     |        |
| 3-9     |        |
| 3-10    |        |

Please copy the above answer table to your answer sheet.

The following program can transfer an infix expression into a postfix expression according to the BODMAS (Brackets, Orders, Division, Multiplication, Addition, and Subtraction) rule that is used for ordering mathematical operations. In this program, suppose that brackets are not used. The operators include "+, -, \*, /", and the operands consist of "a ~ z". The sample inputs of the program are: {"a+b\*c-d/e^f", "a\*b-c/d^e\*f+g\*h"}, and their corresponding outputs are {"abc\*\*def^-", "ab\*cde~/f\*-gh\*+"}. Please trace this program and fill the 3-1~3-10 blanks with correct answers.

```
#include <stdio.h>
#define SIZE 20
typedef enum {false=0, true=1} bool;
bool isEmpty(char stack[], int top) {
    if ( problem 3-1 <=-1) return true;
    else return false;
}
bool push(char stack[], int *topp, char element) {
    stack[ problem 3-2 (*topp)] = element;
    return true;
}
bool pop(char stack[], int *topp, char *element) {
    if (isEmpty(stack, (*topp))) return false;
    (*element) = stack[(problem 3-3)];
    return true;
}
void popAll(char stack[], int *topp, char postfix[],
            int *indexp) {
    while (! problem 3-4 (stack, (*topp))) {
        postfix[( problem 3-5 )++]
            =stack[(problem 3-3)--];
        printf("%c", postfix[(problem 3-5)-1]);
    }
}
bool isOperand(char c) {
    if ((c>='a') problem 3-6 (c<='z')) return true;
    else return false;
}
int opPriority(char c) {
    if (c=='^') return 3;
    else if (c=='*' || c=='/') return 2;
    else if (c=='+' || c=='-') return 1;
    else return 0;
}
bool shouldPopOp(char x, char y) {
    if (opPriority(x) problem 3-7 opPriority(y))
        return true;
    else return false;
}
void operatorProcess(char stack[], int *topp,
                    char postfix[], int *index, char c) {
    char elem;
    bool nonEmpty;
    nonEmpty = pop(stack, topp, &elem);
    while (nonEmpty&&shouldPopOp(elem, c)) {
        postfix[(problem 3-8)] = elem;
        printf("%c", problem 3-8);
        nonEmpty = pop(stack, topp, &elem);
    }
    if ( problem 3-9 ) push(stack, topp, elem);
    push(stack, topp, c);
}
void operandProcess(char postfix[], int *indexp,
                    char c) {
    postfix[(problem 3-10)] = c;
    printf("%c", c);
}
void infixToPostfix(char infix[], char postfix[],
                    int size) {
    int i=0, index=0, top=-1;
    char stack[SIZE];
    for (i=0; i<size; i++) {
        if (isOperand(infix[i])) {
            operandProcess(postfix, &index, infix[i]);
        }
        else {
            operatorProcess(stack, &top, postfix,
                            &index, infix[i]);
        }
    }
    popAll(stack, &top, postfix, &index);
}
void test01() {
    char infix1[] = "a+b*c-d/e^f";
    char infix2[] = "a*b-c/d^e*f+g*h";
    char postfix[30];
    int size = 15;
    infixToPostfix(infix1, postfix, size);
    infixToPostfix(infix2, postfix, size);
}
```

**Problem 4 [24%, each 3%]**

Please trace below program and fill the blanks 4-1~4-5 with correct statements and answer the problems 4-6~4-8.

```

#include <iostream>
using namespace std;
class Compute {
public:
    virtual           problem 4-1          
    virtual void print()=0;
};
class Data {
public:
    Data(int size) {
        _size = size;
        _score = new int[_size];
        for (int i=0; i<_size; i++) _score[i] =
            80+2*i;
    }
    int mean() {
        int mValue=0;
        for (int i=0; i<_size; i++)
            mValue+=_score[i];
        return (          problem 4-2          );
    }
private:
    int           problem 4-3          
    int _size;
};
template <class T>
class AbstractData {
public:
    AbstractData(int size,           problem 4-4          ) {
        _size = size;
        _score = new T[_size];
        for (int i=0; i<size; i++)
            _score[i]=score[i];
    }
    T mean() {
                  problem 4-5          
        for (int i=0; i<_size; i++)
            mValue+=_score[i];
        return (mValue/_size);
    }
private:
    T *_score;
    int _size;
};

class Group :public Compute {
public:
    Group(int size, int score[]) {
        data1= new AbstractData<int>(size,
score);
        data2= new Data(size);
    }
    bool lessThan() {
        if (data1->mean()>data2->mean())
            return false;
        else
            return true;
    }
    void print() {
        cout<<"data set 1: "<<
data1->mean()<<endl;
        cout<<"data set 2: "<<
data2->mean()<<endl;
    }
private:
    AbstractData<int> *data1;
    Data *data2;
};

void test05() {
    int size =5;
    int score[] = {90, 80, 85, 88, 86};
    Group g(5, score);
    Data data(8);

// problem 4-6: the output of below cout
    cout<<data.mean()<<endl;

// problem 4-7: the output of below g.print()
    g.print();

// problem 4-8: the output of below cout
    cout<<g.lessThan();
}

int main() {
    test05();
    return 0;
}
    
```

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

Please copy the above answer table to your answer sheet.

**Problem 5 [4%]**

Please copy below class diagram and complete the diagram by showing the class's attributes, operations, and relationships for the classes in the program of Problem 4.



