114CI02

# 國立臺北科技大學 114 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



**注意事項：**

1. 本試題共 五 題，共 100 分。
2. 不必抄題，作答時請將試題題號及答案依照順序寫在答案卷上。
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 [31%] [1-1 ~ 1-3, each 3%] [1-4 ~ 1-14, each 2%]**

(1) Please trace the following Python program and answer the output of each "print" statement for problems 1-1 ~ 1-6.

```
01  def getNum(n: int, m: int):
02      if n<=0: return 0
03      elif n%m==0: return n + getNum(n-m-1, m)
04      else: return getNum(n-1, m)
05
06  def getIndex(index:int, i: int):
07      if index & (1 << i) > 0: return True
08      return False
09
10  def getSubset(data: str, index: int, subset=''):
11      for i in range(len(data)):
12          if getIndex(index, i)==True:
13              subset += data[i]
14      return subset
15
16  def getData(data: list, key: tuple):
17      r = key(data)
18      return r[0] + r[1]
19
20  def getDict(ns, ss, name_p='', students={}):
21      for i in range(len(ns)):
22          students[ns[i]] = ss[i]
23      for key, value in students.items():
24          if value>=60: name_p += key
25      return name_p
26
27  cnt, data, x = 3, '', [7, 6, 5, 4, 3]
28  names, scores = ['O', 'M', 'K', 'J'], [90, 55, 100, 50]
29  print(getData(x, key=lambda k:(k[1], k[3])))          # problem 1-1
30  print(getNum(20, 10)+getNum(15, 5))                   # problem 1-2
31  for i in range(10):
32      if getIndex(6, i): cnt += 1
33  print(cnt)                                            # problem 1-3
34  for index in range(1, 10):
35      subset = getSubset('csie', index)
36      if len(subset)==3: data += subset
37  print(data)                                           # problem 1-4
38  print(getDict(names, scores))                         # problem 1-5
39  print(getIndex(5, 5))                                 # problem 1-6
```

(2) Suppose the output of following Python program is "1A3B 1A1B, 0A0B -1A-1B, False True, False 1623, 1023 0, 14", please trace the code and fill the blanks problems 1-7~ 1-14 with correct statements.

```
01  #To compare X and Y with position and symbol
02  def matchAB(X: str, Y: str, ab=0):
03      if (len(X) != len(Y)):
04          ab =_____                                     # problem 1-7
05          return '%dA%dB' %(ab, ab)
06      for i in range(len(X)):
07          if X[i]==Y[i]: ab += _____                    # problem 1-8
08          elif X[i] in _____: ab += 1                   # problem 1-9
09      return '%dA%dB' %(ab//10, ab%10)
10
11  # To transfer decimal 'num' into 'base' base.
12  def decimal_to_base(num: int, base: int, ans=''):
13      encoding = '0123456789ABCDEF'
14      if num == 0: return encoding[0]
15      indices = []
16      while num>0:
17          x = num _____ base                            # problem 1-10
18          indices.insert(0, x)                          # insert at position 0
19          num = num _____base                           # problem 1-11
20      for i in indices:
21          ans += encoding[i]
22      return ans
23
24  def printNum(start: int, stop:int, cnt=0):
25      for i in range(start, stop):
26          encode = decimal_to_base(i, 16)
27          if isDuplicate(encode, 4)==True:
28              cnt += _____                              # problem 1-12
29      return cnt
30
31  # The elements of data cannot be duplicated, and the length must be equal to n.
32  def isDuplicate(data: str, n: int):
33      if len(data) != n: return _____                   # problem 1-13
34      for d in data:
35          if data.count(d) > 1: return False
36      return _____                                      # problem 1-14
37
38  print(matchAB('AB10', '1B0A'), matchAB('9B1F', '1B0A') , end=', ')
39  print(matchAB('123F', '567D'), matchAB('B10', '1B0A') , end=', ')
40  print(isDuplicate('AA10', 4), isDuplicate('A10F', 4), end=', ')
41  print(isDuplicate('AA0F', 5), decimal_to_base(1236, 9), end=', ')
42  print(decimal_to_base(4131, 16), decimal_to_base(0, 16), end=', ')
43  print(printNum(4124, 4138))
```

注意：背面尚有試題

**Problem 2 [15%] [2-1 ~ 2-7 each 2%, 2-8 1%]**
Please trace the following C program and fill the blanks problems 2-1 ~ 2-5 with correct statements, and answer the output of each "printf" statement for problems 2-6 ~ 2-8.

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  #define SIZE 26
04  int comp(char * s, char *d) {
05        if ((*s=='\0') && (*d=='\0')) return 0;
06        else if ((*s=='\0') || (*s<*d)) return -1;
07        else if ((*d=='\0') || (*s>*d)) return 1;
08        else return comp(s+1, d+1);
09  }
10  // To initialize the statistic table of alphabet letters - "alpha_table".
11  void init_alpha_table(int * alpha_table){
12        for(int i = 0; i < SIZE; i++) alpha_table[i] = 0;
13  }
14  // To covert the characters of string "buf" into lowercase.
15  void lowercase(char* buf){
16        for(int i = 0; buf[i]!='\0'; i++)
17              if ('A' <= buf[i]_______)                          /* Problem 2-1 */
18                    buf[i] = 'a' + buf[i] -_________;            /* Problem 2-2 */
19  }
20  // To count the number of characters c in the string "data".
21  int alpha_count(char data[], char c){
22        int total = 0;
23        for(int i = 0; i < SIZE; i++)
24              if(data[i] == c) _________ = 1;                    /* Problem 2-3 */
25        return total;
26  }
27  //Count the frequency of alphabet letters in the string "buf", and store in "alpha_table" tabke.
28  void count_Freq(char buf[], int alpha_table[]) {
29        for (int i = 0; i < SIZE; i++)
30              alpha_table[i] = alpha_count(buf, ('a'_______));   /* Problem 2-4 */
31  }
32  // To get the value stored in the "alpha_table" table, the value is the number of "specific_c".
33  int get_num_alpha_table(int* alpha_table, char specific_c){
34        int index = specific_c_______'a';                        /* Problem 2-5 */
35        return alpha_table[index];
36  }
37  int main(void){
38        int alpha_table[SIZE];
39        char buf[] = "2024NtUtCsie";
40        init_alpha_table(alpha_table);
41        printf("%d\n", alpha_count(buf, '2'));                   // Output: 2
42        lowercase(buf);
43        printf("%s\n", buf);                                     // Output: "2024ntutcsie"
44        count_Freq(buf, alpha_table);
45        printf("%d\n", get_num_alpha_table(alpha_table, 't'));   // Output: 2
46        printf("%d\n", get_num_alpha_table(alpha_table, 'n'));   // Output: 1
47        printf("%d\n", get_num_alpha_table(alpha_table, 'c'));   // Output: 1
48        printf("%d\n", alpha_count("book_is_good", 'o'));        /* Problem 2-6 */
49        printf("%d, %d\n", comp("FO", "FOO"), comp("FO", "FO")); /* Problem 2-7 */
50        printf("%d, %d\n", comp("FOOD", "FOOAD"), comp("", "")); /* Problem 2-8 */
51  }
```

**Problem 3 [20%] [3-1 ~ 3-2 each 3%, 3-3 ~ 3-9 each 2%]**
The following C program can calculate the addition of two polynomials. Please trace the program and fill the blanks problems 3-1 ~ 3-9 with correct statements.

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  typedef _______ node_s {                                       /* Problem 3-1 */
04        int coef;
05        int exp;
06        struct node_s * next;
07  } node_t;
08  typedef node_t * nodep_t;
09  int count(int num) {
10        int sum = 0;
11        while (num>0) {
12              sum += num_________;                               /* Problem 3-2 */
13              num = num/10;
14        }
15        return sum;
16  }
17  nodep_t makeNode(int coef, int exp) {
18        nodep_t newNode = (nodep_t) malloc(_____________);       /* Problem 3-3 */
19        newNode->next = NULL;
20        newNode->coef = coef;
21        newNode->exp = exp;
22  }
23  void print(nodep_t x) {
24         while (x!=NULL) {
25              if (x->coef>0) printf("+");
26              if (x->exp ==0) printf(" %d ", x->coef);
27              else if (x->exp==1) printf("%d x ", x->coef);
28              else     printf("%d x^%d ", x->coef, x->exp);
29              x = x->next;
30         }
31        printf("\n");
32  }
33  nodep_t addNode(nodep_t root, nodep_t newNode) {
34        nodep_t current = root;
35        if (current == NULL) return newNode;
36        while (current->next != NULL)
37              current = current->next;
38        current->next = _________________;                       /* Problem 3-4 */
39        return root;
40  }
```

```
41  nodep_t add(nodep_t x, nodep_t y) {
42      nodep_t root = NULL, newNode = NULL, current = NULL;
43      while ((x!= NULL) && (y!=NULL)) {
44          if (x->exp == y->exp) {
45              newNode = makeNode( _____, x->exp);          /* Problem 3-5 */
46              x=x->next; y = y->next;
47          }
48          else if (x->exp > y->exp) {
49              newNode = makeNode(x->coef, x->exp);
50              x= _____;                                    /* Problem 3-6 */
51          }
52          else {
53              newNode = makeNode( _____, y->exp);          /* Problem 3-7 */
54              y = y->next;
55          }
56          root = addNode( _____, newNode);                 /* Problem 3-8 */
57      }
58  }
59  nodep_t make_polynomial(nodep_t root, int data[], int n) {
60      for (int i=0; i<=n; i++)
61          if (data[i]____0) root=addNode(root, makeNode(data[i],n-i)); /* Problem 3-9 */
62      return root;
63  }
64  int main() {
65      nodep_t x = NULL, y=NULL, root=NULL, z=NULL;
66      int data_x[] =   {2, 0, 3, -4, 5};          //      2x^4 + 0x^3 + 3x^2 - 4x + 5
67      int data_y[] = {1, 0, 0, 2, 0, 1};          //1x^5 + 0x^4 + 0x^3 + 2x^2 + 0x + 1
68      x = make_polynomial(x, data_x, 4);
69      y = make_polynomial(y, data_y, 5);
70      print(x);                                   //Output: +2 x^4 +3 x^2 -4 x + 5
71      print(y);                                   //Output: +1 x^5 +2 x^2 + 1
72      z = add(x, y);
73      print(z);                                   //Output: +1 x^5 +2 x^4 +5 x^2 -4 x + 6
74      printf("%d, %d", count(98765), count(1357)); //Output: 35, 16
75      return 0;
76  }
```

**Problem 4 [17%] [4-1 ~ 4-6 each 2%, 4-7 ~ 4-11 each 1%]**

Please trace the following C++ program and fill the blanks **problems 4-1 ~ 4-7** with correct statements, and answer the output of each "cout" statement for **problems 4-8 ~ 4-11**.

```
01  #include <iostream>
02  #include <vector>
03  using namespace std;
04  class Product {
05    protected:
06      int value;
07    public:
08      Product(int v): value(v) {value+=10;}
09      /* Problem 1-1: Fill in the code for a pure virtual function declaration */
10      _____;                                          // Problem 4-1
11      int getValue() {return computeTax() + _____;}   // Problem 4-2
12  };
13  class Food: public Product {
14    public:
15      Food(int v): Product(v) {value += 10;}
16      int computeTax() override { return value * _____;}   // Problem 4-3 (Integer)
17  };
18  class House: public Product {
19    public:
20      House(int v): Product(v) {}
21      int computeTax() override { return value*0.2 + _____;}   // Problem 4-4 (Integer)
22  };
23  class Car: public Product {
24    public:
25      Car(int v): Product(v) {}
26      int computeTax() override { return value%80 - _____; }   // Problem 4-5 (Integer)
27  };
28  int sumOfComputes(const vector<Product*>_____) {          // Problem 4-6
29      int sum = 0;
30      for (const auto& obj: objects)
31          sum += obj_____;                                  // Problem 4-7
32      return sum;
33  }
34  int main() {
35      Food f1(10), f2(20);
36      House h1(20), h2(30);
37      Car   c1(30), c2(10);
38      vector<Product*> objects = {&f1, &h1, &c1};
39      cout << f1.computeTax() << endl;          // Output is 60
40      cout << h1.computeTax() << endl;          // Output is 16
41      cout << c1.computeTax() << endl;          // Output is 22
42      cout << f1.getValue() << endl;            // Output is 90
43      cout << sumOfComputes(objects) << endl;   // Output is 98
44      cout << f2.computeTax() << endl;                  // Problem 4-8
45      cout << h2.computeTax() << endl;                  // Problem 4-9
46      cout << c2.computeTax() << endl;                  // Problem 4-10
47      cout << f2.getValue() << endl;                    // Problem 4-11
48  }
```

注意：背面尚有試題

**Problem 5 [17%] [5-1 ~ 5-6 each 2%, 5-7 ~ 5-11 each 1%]**
Please trace the following C++ program and fill the blanks problems 5-1 ~ 5-3 with correct statements, and answer the output of each "cout" statement for problems 5-4 ~ 5-11

```
01  #include <iostream>
02  #include <vector>
03  #include <string>
04  #include <stdexcept>
05  using namespace std;
06  template <class T>
07  class Box {
08    private:
09      T content;
10      _____ boxCount;                                   // Problem 5-1
11    public:
12      Box( _____ c): content(c) { boxCount++; }         // Problem 5-2
13      T getContent() const { return content; }
14      static int getCount() { return boxCount; }
15      ~Box() { boxCount--; }
16  };
17  template <class T>
18  int Box<T>::boxCount = 0;      /* Static member initialization*/
19  class Shape {
20    protected:
21      _____ ;                                           // Problem 5-3
22    public:
23      Shape(string n): name(n) {}
24      virtual string getName() const = 0;
25      virtual ~Shape() {}
26  };
27  class Square : public Shape {
28    private:
29      int side;
30    public:
31      Square(string n, int s): Shape(n), side(s) {}
32      string getName() const override { return "C:" + name; }
33      int getArea() const { return side * side; }
34  };
35  class Rectangle : public Shape {
36    private:
37      int length, width;
38    public:
39      Rectangle(): Shape("d"), length(1), width(1) {}
40      Rectangle(string n, int l, int w): Shape(n), length(l), width(w) {}
41      string getName() const override { return "R:" + name; }
42      double getArea() const { return length * width; }
43  };
```

```
44  int main() {
45      try {
46          Box<Square> sBox1(Square("S1", 3));
47          Box<Square> sBox2(Square("S2", 2));
48          Box<Rectangle> rBox(Rectangle("R1", 4, 5));
49          Rectangle rect;
50
51          cout << "RName: "<< rect.getName()<<endl;                  // Problem 5-4
52          cout << "RArea: "<< rect.getArea()<<endl;                  // Problem 5-5
53          cout << "SArea: "<< sBox1.getContent().getArea()<<endl;    // Problem 5-6
54          cout << "RArea: "<< rBox.getContent().getArea()<< endl;    // Problem 5-7
55          cout << "sBoxes: "<< Box<Square>::getCount()<< endl;       // Problem 5-8
56          cout << "rBoxes: "<< Box<Rectangle>::getCount()<< endl;    // Problem 5-9
57          Square square = sBox1.getContent();
58          Rectangle rectangle = rBox.getContent();
59          vector<Shape*> shapes = {&square, &rectangle};
60          for (auto shape:shapes) cout<<shape->getName()<<endl;      // Problem 5-10
61          cout << shapes.at(2)->getName() << endl;
62      } catch (const exception& e) {
63          cout << "E: " <<"Out of Index"<< endl;                     // Problem 5-11
64      }
65  }
```