

考試別：司法人員
等 別：三等考試
類 科 組：檢察事務官電子資訊組
科 目：程式語言
考試時間：2 小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、針對以下 C 程式碼，請先說明執行該程式之輸出，以及程式中函式 left()、right() 之運作功能；並請說明在 testGen() 測試 left()、right() 過程中，有那幾行指令未被充分測試。(25 分)

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 char* left(char* data, int len, const char symbol) {
05     int dLen = strlen(data);
06     char* result = (char*)malloc(dLen + 1);
07     if (len > dLen) len = dLen;
08     strcpy(result, data);
09     for (int i = len; i < dLen; i++) result[i] = symbol;
10     return result;
11 }
12 char* right(char* data, int len, const char symbol) {
13     int dLen = strlen(data);
14     char* result = (char*)malloc(dLen + 1);
15     if (len > dLen) len = dLen;
16     strcpy(result, data);
17     for (int i = (dLen-len); i < dLen; i++) result[i] = symbol;
18     return result;
19 }
20 void testGen() {
21     char str[] = "abcdef";
22     char* left1 = left(str, 4, '@');
23     char* left2 = left(str, 8, '@');
24     char* right1 = right(str, 4, '@');
25     printf("%s\n", left1);
26     printf("%s\n", left2);
27     printf("%s\n", right1);
28     free(left1);
29     free(left2);
30     free(right1);
31 }
32 int main() {
33     testGen();
34     return 0;
35 }
```

二、以下 Java 程式碼計算人類 (Human) 進行不同運動 (Exercise) 消耗的卡路里量。請完成統一塑模語言 UML 圖；並說明程式執行後的輸出，以及動態多型 (Dynamic Polymorphism) 指令所在之程式碼行數與其構成之原因。(25 分)

```

01  abstract class Exercise { // 運動
02      public Exercise(String name, int cal) {
03          this.name = name;
04          this.calorie = cal;
05      }
06      abstract public void computeCal(int x);
07      public String show(int c) {
08          String result = name+":";
09          computeCal(c);
10          result += calorie;
11          return result;
12      }
13      private String name;
14      protected int calorie; // 紀錄卡路里
15  }
16  class Running extends Exercise { // 跑步運動
17      public Running(String name, int cal, int dis) {
18          super(name, cal);
19          distance = dis;
20      }
21      public void computeCal(int time) { calorie += distance*(10-time); }
22      private int distance;
23  }
24  class Climbing extends Exercise { // 登山運動
25      public Climbing(String name, int cal, int lev) {
26          super(name, cal);
27          level = lev*2;
28      }
29      public void computeCal(int height) { calorie += height*level*20; }
30      private int level;
31  }
32  class Human {
33      public void setExercise(Exercise e) {exercise = e;}
34      public String getCal() { return exercise.show(2);}
35      private Exercise exercise;
36  }
37  public class ExerciseTest {
38      public static void main(String[] args) {
39          Exercise [] exs = {new Running("marathon", 12, 26), // 馬拉松
40                          new Running("endurance", 10, 5), // 長跑
41                          new Climbing("Yushan", 30, 4)}; // 登玉山
42          Human h = new Human();
43          for (Exercise e: exs) {
44              h.setExercise(e);
45              System.out.println(h.getCal());
46          }
47      }
48  }

```

Exercise

Running
distance

Climbing
level

Human

三、以下 Python 程式碼，可以解析下列 BNF 規則。請先寫出 BNF 規則中 (I) ~ (V)；並說明程式執行後之輸出。(25 分)

<assign>	<id> = <_(I)>
<id>	_(II)_
<expr>	<expr>_(III)<expr> <expr>_(IV)<expr> <_(V)>

01	def id_(s, i):
02	if i < len(s) and s[i] in ('A', 'B', 'C'):
03	return i + 1, True
04	return i, False
05	
06	def expr(s, i):
07	i, ok = id_(s, i)
08	if not ok: return i, False
09	while i + 1 < len(s) and s[i] in ('+', '*'):
10	i, ok = id_(s, i+1)
11	if not ok: raise Exception('id error')
12	return i, True
13	
14	def assign(s, i):
15	i, ok = id_(s, i)
16	if not ok or i >= len(s): raise Exception('id error')
17	if s[i] != '=': raise Exception('not =')
18	i, ok = expr(s, i+1)
19	if not ok: raise Exception("expr error")
20	if i != len(s): raise Exception('end or expr error')
21	print('ok')
22	
23	def test(code):
24	try: assign(code, 0)
25	except Exception as e: print(e)
26	
27	test('B=C*B+A*A')
28	test('B=C*B+D*A')
29	test('B=C*B+A*')
30	test('A-C*D+A++A')
31	test('A=C*D+A++A')

- 四、臺灣智慧金融公司設計一個關聯式資料庫 (Relational database)，內有兩個關聯 (Relation)：工程師 Engineer[編號 (整數、Primary key)、姓名 (字串)、電話 (字串)、專長 (字串)]、助理 Assistant[編號 (整數、Primary key)、姓名 (字串)、薪水 (整數)、工程師編號 (Foreign key)]。
- 請寫出 SQL 指令，查詢出工程師編號小於等於 50 且其助理薪水大於 30000 的工程師姓名與專長，並造出工程師專長為「hack」相關之姓名與協助的助理姓名與電話的 View。另請說明何謂 View，以及使用 View 的優點。(25 分)