計算機程式設計 113 學年度第 2 學期小考 1 試題

系級：_____ 學號：_____ 姓名：_____

All programs have "#include <stdio.h>".

【Each space 3%.】

1. Fill in the space to complete the code.

(1)_____%d_____ (2)_____ %c_____ (3)_____ %.3f_____

Output	10 A 10.000

```
01  int main() {
02      int a = 10;
03      char b = 'A';
04      float c = 10;
05      printf("___(1)___\n", a);
06      printf("___(2)___\n", b);
07      printf("___(3)___\n", c);
08      return 0;
09  }
```

2. Fill the space to complete the usage of variadic arguments.

(1)____va_start____ (2)___ va_arg____ (3)_____ va_end____

Output	The sum is: 10

```
01  #include <stdarg.h>
02  int sum(int count, ...) {
03      va_list args;
04      int total = 0;
05      ___(1)___(args, count);
06      for (int i = 0; i < count; i++) {
07          total += ___(2)___(args, int);
08      }
09      ___(3)___(args);
10      return total;
11  }
12  int main() {
13      int result = sum(4, 1, 2, 3, 4);
14      printf("The sum is: %d\n", result);
15      return 0;
16  }
```

3. Output after executing the following code.

(1)___000121____ (2)_____01_____ (3)_____00_____

```
01  void switch_case_func(int z) {
02      int x = 0, y = 0;
03      z %= 5;
04      switch (z) {
05          case 0:
06              printf("%d%d", x, y++);
07          case 1:
08              printf("%d%d", x++, y);
09          case 2:
10              printf("%d%d", ++x, y);
11              break;
12          case 3:
13              printf("%d%d", x, ++y);
14              break;
15          default:
16              printf("%d%d", x, y);
17              break;
18      }
19  }
20  int main() {
21      switch_case_func(15); //(1)
22      printf("\n");
23      switch_case_func(18); //(2)
24      printf("\n");
25      switch_case_func(19); //(3)
26      printf("\n");
27      return 0;
28  }
```

4. What are the three types of program structures?

(1) 循序結構(Sequential Structure) (2) 選擇結構(Selection Structure) (3) 重複結構(Repetition Structure)

5. Output after executing the following code.

(1)_____9_____ (2)_____63_____ (3)____54_____

```
01  int main() {
02      int a = 0x65;
03      int b = 057;
04
05      a = a >> 2;
06
07      printf("%d\n", a & b); // (1)
08      printf("%d\n", a | b);  // (2)
09      printf("%d\n", a ^ b);  // (3)
10
11      return 0;
12  }
```

6. Output after executing the following code.

(1)_____53_____ (2)_____56_____ (3)_____9_____

```
01  char f(int x) {
02      return x + '0';
03  }
04
05  int main() {
06      printf("%d\n", f(1)); // output: 49
07      printf("%d\n", f(5)); // (1)
08      printf("%d\n", f(8)); // (2)
09      printf("%c\n", f(9)); // (3)
10  }
```

7. In the html report generated by the coverage package, describe the meaning of the green, yellow, and red color.

Green：_____指令有被充分測試到_____

Yellow：____條件判斷 T/F 指令，部分被測試到_____

Red：_____指令沒被測試到_____

8. Please describe the two definitions.

Line coverage：

用來衡量測試過程中已執行的可執行程式碼語句的百分比。它指示測試案例覆蓋了多少個單獨的程式碼語句。

Branch coverage：

用來衡量測試期間已執行的程式碼中分支的百分比。側重於程序的控制流程，旨在確保所有可能的分支都經過測試。

9. (1)___(A)(E)____ (2)_____(D)____ (3)____ H____

(1) Choose the following instructions to achieve 100% "Branch coverage". (Multiple choice)

(2) There is only one option, which makes it impossible to achieve 100% “Line coverage.” What is that option?

(A) add (-4, 4), add (5, -5), add (3, 4) , add(8, 8), add(10,13)

(B) add (-1, 1), add (0, 0), add (2, 4) , add(7, 9), add(12,13)

(C) add (-1, -1), add (-3, 3), add (2, 4) , add(5, 5), add(10,13)

(D) add (2, 2), add (4, 5), add (6, 8) , add(10, 10), add(-1,-1)

(E) add (-3, -5), add (3, -4), add (0, 0) , add(5, 5), add(15,15)

(3) Also, what is the return value of the uncovered line of code?

```
01  char add(int a, int b) {
02      if (a < 0 || b < 0) {
03          return "Z";
04      }
05      int sum = a + b;
06      if (sum < 10) {
07          return "L";
08      } else if (sum <= 20) {
09          return "N";
10      } else {
11          return "H";
12      }
13  }
```

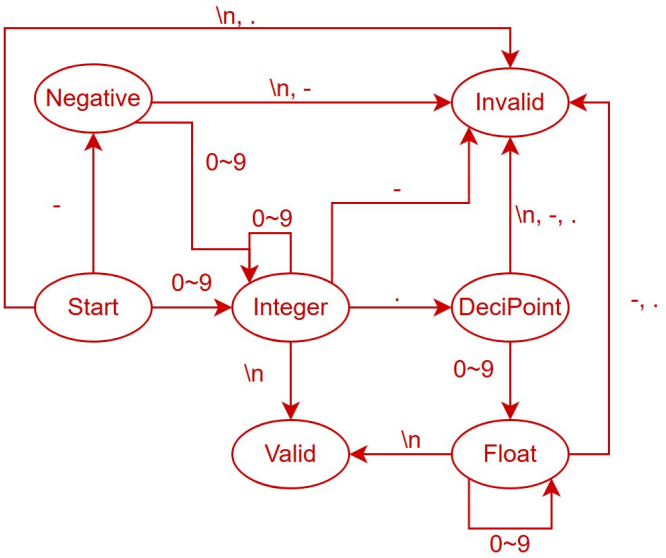
10. Output after executing the following code.

(1)_____6_____ (2)_____7_____ (3)_____8_____

```
01  int g(int x);
02  int f(int y);
03
04  int f(int x) {
05      if (x <= 0) return 1;
06      return g(x - 2) + 1;
07  }
08  int g(int y) {
09      if (y <= 0) return 3;
10      return f(y) + 1;
11  }
12  int main() {
13      printf("%d\n", f(3));      //(1)
14      printf("%d\n", g(3));      //(2)
15      printf("%d\n", f(g(2)));  //(3)
16      return 0;
17  }
```

11. Please analyze the following code and draw a “State Transition Diagram” to describe the code sequence. Notice that keys can only be numbers, dashes, dots, and ‘\n’.

```
01  int getState(int state, char key) {
02      if (state == START && key >= '0' && key <= '9') {
03          return INTEGER;
04      }
05      else if (state == START && key == '-') {
06          return NEG;
07      }
08      else if (state == NEG && key >= '0' && key <= '9')    {
09          return INTEGER;
10      }
11      else if (state == INTEGER && key >= '0' && key <= '9') {
12          return INTEGER;
13      }
14      else if (state == INTEGER && key == '.') {
15          return DECI_POINT;
16      }
17      else if (state == DECI_POINT && key >= '0' && key <= '9') {
18          return FLOAT;
19      }
20      else if (state == FLOAT && key >= '0' && key <= '9') {
21          return FLOAT;
22      }
23      else if ((state == INTEGER || state == FLOAT) && key == '\n'
24          return VALID_NUMBER;
25      }
26      else {
27          return INVALID;
28      }
29  }
```



12. According to the problem description, answer the output for the following test cases.

(1)___X Win_____ (2)_____ Tie_____ (3)____ Y Win_____

The user needs to input two cards for both Player X and Player Y in sequence, and then sum up the points. If the total score exceeds 10.5, the score is reset to 0. If Player X has a higher score than Player Y, output "X Win"; if Player Y has a higher score than Player X, output "Y Win"; if both players have the same score, output "Tie."

```
01  Poker(4, 5, 3, J);   //(1)
02  Poker(5, 7, 6, 6);  //(2)
03  Poker(1, 3, 2, 4);  //(3)
```

13. Complete the program to calculate the total number of infected people after a given number of days using recursion. Starting with 1 infected person on day 0, each person infects a fixed number daily. Fill in the blanks and determine the value of final totalInfected.

(1)___days-1____ (2)_____rate_____ (3)_____27_____

```
01  int virusSpread(int days, int rate) {
02      if (days == 0) {
03          return 1;
04      }
05
06      int previousTotal = virusSpread(___(1)___,___(2)___);
07      return previousTotal + (previousTotal * rate);
08  }
09
10  int main() {
11      int days = 3;
12      int rate = 2;
13
14      int totalInfected = virusSpread(days, rate);
15
16      printf("%d\n", totalInfected); //(3)
17      return 0;
18  }
```

14. Output after executing the following code.

(1)_____13_____ (2)_____55_____ (3)_____233_____

```
01  int Fibonacci(int num) {
02      int a = 1, b = 1;
03      for (int i = 2; i < num; i++) {
04          int temp = a + b;
05          a = b;
06          b = temp;
07      }
08      return b;
09  }
10  int main() {
11      printf("%d\n", Fibonacci(7));   // (1)
12      printf("%d\n", Fibonacci(10)); // (2)
13      printf("%d\n", Fibonacci(13)); // (3)
14  }
```

15. For this course, please point the more confusing section. Write down the problem and how to improve it. (30 words or more will be scored)