

程設

CS 1355 Fall 2008

Midterm Exam

1.b2

- This is a closed book, closed notes exam. You have 50 minutes. (考試中不可翻閱課本及筆記. 考試時間為50分鐘)
- You must sign your name to certify you are the actual person taking the exam. (考生必須簽名證實是本尊赴考)
- Mark your answers clearly by filling in the bubbles on the answer sheet, unless otherwise instructed. (選擇題以在答案紙上塗滿圓圈作答)
- There are 25 questions. (總共有25題) Some questions are more time consuming than others. Read the questions carefully. You might want to answer the easier questions first and come back to the more time consuming ones. (題目困難度不一, 所以不要只顧難題而忽略了簡單題)
- Turn off all your electronic devices. No calculators or dictionaries allowed. (一切電子用品都需被關機. 不可使用計算機或字典)
- Line numbers may also be displayed for convenience, but they are not part of the program. (程式前可能有標註行例數, 以便問答, 但並非程式的一部份。)
- Some multiple-choice answers may go across pages. (題目及答案可能跨頁)

Answer Sheet

5

Q#	a	b	c	d	e
1	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
5	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Q#	a	b	c	d	e
11	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
12	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
15	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
17	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
18	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Q#	a	b	c	d	e
21	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
24	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
25	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

80

1. double is a (an)

- a) identifier b) literal c) keyword
d) operator e) shell

2. + is a (an)

- a) literal b) identifier c) keyword
d) shell e) operator

3. What is the evaluation ordering of the expression

((a + (b * c)) > d) || (e - (f / g)))

- a) (a + (b * (c > d))) || (e - (f / g))
b) (((a + b) * c) > d) || ((e - f) / g)
c) ((((((a + b) * c) > d) || e) - f) / g)
d) ((a + (b * c)) > d) || (e - (f / g))
e) (a + b) * ((c > (d || e)) - (f / g))

4. What is the output of the statement

printf("%012d", 1234); ? Note that we use the _ character to show each white space. (以下利用 _ 來表示印出來的空白)

- a) 0000000000001234 b) _____1234
c) 0000001234 d) 1234_____
e) 000000001234

5. Assume `int x=1; int y[5]={0};` Which of the following expressions has no L-value?

- a) y[x] b) y c) y[0] d) x e) y[y[x]]

6. Which of the following is not equivalent to the others? (以下哪一個程式功能與其他不同)

a)

```
if (x >= 90) {  
    printf("A");  
}  
if (90 > x && x >= 80) {  
    printf("B");  
}  
if (80 > x && x >= 70) {  
    printf("C");  
} else {  
    printf("F");  
}
```

90 → A.F
80 → B.F
70 → C
60 → F

b)

```
if (x >= 90) {  
    printf("A");  
} else if (x >= 80) {  
    printf("B");  
} else if (x >= 70) {  
    printf("C");  
} else {  
    printf("F");  
}
```

90 → A

80 → B

70 → C

60 → F

c)

```
if (x >= 90) {  
    printf("A");  
} else {  
    if (x >= 80) {  
        printf("B");  
    } else {  
        if (x >= 70) {  
            printf("C");  
        } else {  
            printf("F");  
        }  
    }  
}
```

d)

```
if (x >= 90) {  
    printf("A");  
} else if (90 > x && x >= 80) {  
    printf("B");  
} else if (80 > x && x >= 70) {  
    printf("C");  
} else {  
    printf("F");  
}
```

e)

```
if (x >= 90) {  
    printf("A");  
} else if (x < 90 && x >= 80) {  
    printf("B");  
} else if (x < 80 && x >= 70) {  
    printf("C");  
} else {  
    printf("F");  
}
```

7. Suppose you want to use a while-loop to print numbers 1 to 10. Here is a program:

```
1 int count = 0;  
2 while (count < 10) {  
3     printf("%d\n", count);  
4     count = count + 1;  
5 }
```

but this program does not work correctly. What change will make it print correctly?

a) Exchange line 3 and line 4

b) Actually, this program works correctly.

c) Change line 4 to `count += 1;`

d) Change line 2 to `while (count <= 10) {`

e) Change line 1 to `count = 1;`

The next two questions refer to the following program. Suppose you want to compute the average of grades by prompting the user for one grade at a time, and the user signifies the end by typing -1. Given the following *incorrect* program:

```
1 int total = 0;
2 int count = 0;
3 int grade = 0;
4 while (grade != -1) {
5     total = total + grade;
6     count = count + 1;
7     printf("enter grade ");
8     scanf("%d", &grade);
9 }
10 printf("average=%6.2f", (total/count));
```

8. What is one problem with this program? (這個程式出的問題包括以下哪一個?)

a) Line 4 should be changed to (第四行應該改成)

`while (grade == -1) {`

b) It does *not* correctly compute the average, because the count is one larger than the actual count. (演算出來的average不正確,因為count比實際個數多出一個)

c) It does *not* correctly compute the total, because line 5 incorrectly adds an extra grade the first time you enter the while-loop. (total算錯了,因為第一次跳入while-loop時,第五行多加了一個grade)

d) It will enter an infinite loop unless line 3 is changed to (除非第三行照以下改,否則永遠跳不出while-loop)

```
int grade;
printf("enter grade");
scanf("%d", &grade);
```

e) It never enters the while-loop since the condition on line 4 is never true. (因為第四行的condition不可能變true所以永遠跳不進去while loop裡面)

9. What is a problem on line 10? (第十行出了什麼問題?)

a) If every student gets 100 then the `%6.2f` formatting string will start truncating (裁截) digits on the left and show an average of `00.00` on its output.

b) It should say `(total, count)` instead of `(total / count)`.

c) It performs an integer division rather than a floating-point division.

d) The statement should be indented (對齊) the same way as line 8.

e) The string `%6.2f` cannot be a legal format string unless the `6.2` part is removed.

10. Given the code fragment:

```
1 int x = 10;
2 float y = x / 4;
```

$$10 \div 4 = 2 \dots 2$$

What is the value of y after line 2 is executed?

a) 10.0 b) 4.0 c) 2.5 d) 2.0 e) 3.0

11. Given the code

```
for (i = 11; i > 1; --i) {
    printf("%d ", i);
}
```

what will it do?

a) It prints 11 10 9 8 7 6 5 4 3 2

b) It prints 11 10 9 8 7 6 5 4 3 2 1

c) It prints nothing.

d) It prints 10 9 8 7 6 5 4 3 2 1

e) It prints 11

12. Consider the following program, which computes the average of all prompted scores up until the user types in `-1`, and prints a message if the class average is over 60:

```
1 float total = 0.0;
2 int a, count = 0;
3 while (1) {
4     printf("score?");
5     scanf("%d", &a);
6     if (a == -1) break;
7     total += a;
8     count += 1;
9 }
10 if (count > 0 && total/count > 60) {
11     printf("class average > 60\n");
12 }
```

Is there a potential problem on line 10 if count equals zero?

a) No, because line 6 exited the while loop with a `break`, lines 10 to 12 are never executed.

b) Yes, because both sides of the && operator must be completely evaluated, total/count will have a divide-by-zero error if count equals zero.

c) No, because of short-circuit evaluation, total/count will not be executed if count equals zero.

d) No, because you never get an error when you divide a floating point number by an integer.

e) Yes, unless if is replaced with else if.

13. What is the equivalent statement to the following?

`int x = 2 < 3 == 4 > 5;`

a) This is not a syntactically valid statement in C.

b) `int x = 2 < (3 == 4) > 5;`

c) `int x = (2 < (3 == (4 > 5)))`;

d) `int x = (2 < 3) == (4 > 5);`

e) `int x = (((2 < 3) == 4) > 5);`

14. Given the following function:

```
1 int Smaller(int A, int B) {
2   if (A < B) return A;
3   else return B;
4 }
```

What is the value of x after the line

`float x = Smaller(1.7, 3.4);`?

a) 2.0 b) 1.7 c) 3.4 d) 3.0 e) 1.0

The next three questions refer to the following function definition followed by the program to compute the average score:

```
1 #include <stdio.h>
2 void getScore(int score) {
3   int x;
4   printf("enter score:");
5   scanf("%d", &x);
6   score = x;
7 }
8 int main() {
9   int x = 0;
10  getScore(x);
11  printf("score is %d\n", x);
12 }
```

15. Assume you type in 75 when the program prompts you for the score. What does the program print on line 11?

a) score is 0

b) Actually this program will not compile due to an attempt on line 10 to modify the local variable x of the getScore function.

c) score is %d\n

d) score is 75

e) score is %d

16. Suppose you want to assign the letter grade according to the following table:

Score	letter
90-100	A
80-89	B
70-79	C
60-69	D
<60	F

Which of the following functions does NOT correctly perform the conversion? You may assume that scores are in the range of 0...100.

```
1 char scoreToLetter(int score) {
2   if (grade >= 90) score = 'A';
3   else if (grade >= 80) score = 'B';
4   else if (grade >= 70) score = 'C';
5   else if (grade >= 60) score = 'D';
6   else score = 'F';
7   return score;
8 }
```

```
1 char scoreToLetter(int score) {
2   switch (grade/10) {
3     case 6: return 'D';
4     case 7: return 'C';
5     case 8: return 'B';
6     case 9: case 10: return 'A';
7     default: return 'F';
8   }
9 }
```

```
1 char scoreToLetter(int score) {
2   return "FFFFFFDCBAA"[score / 10];
3 }
```

```
1 char scoreToLetter(int score) {
2   char letter = 'F';
3   if (grade >= 60) letter = 'D';
4   if (grade >= 70) letter = 'C';
5   if (grade >= 80) letter = 'B';
6   if (grade >= 90) letter = 'A';
7   return letter;
8 }
```

```

1 char scoreToLetter(int score) {
2   if (grade > 90) return 'A';
3   if (grade > 80) return 'B';
4   if (grade > 70) return 'C';
5   if (grade > 60) return 'D';
6   if (grade < 60) return 'F'
7 }

```

17. What is the output of the following code:

```

1 int A = 100;
2 int B = 3;
3 printf("%.3f", (float) {A/B});

```

- a) 33.000 b) 33.333 c) 33.3
d) .333 e) 33

18. Consider the following switch statement:

```

1 #define Z 45
2 int x, y = 23;
3 scanf("%d", &x);
4 switch (x + 25) {
5   case "A":
6     printf("A"); break;
7   case 'h'+ 'e'+ 'l'+ 'l'+ 'o':
8     printf("hello"); break;
9   case 23:
10    printf("%d", 23); break;
11  case 19.5:
12    printf("%f", 19.5); break;
13  case y % 4:
14    printf("y mod 4"); break;
15  case Z * 3:
16    printf("135"); break;
17  default:
18    break;
19 }

```

Does line 4 contain a valid expression for the switch statement?

- a) yes, any integer expression is valid
b) no, it must be a constant char expression
~~c) no, because x+25 has no R-value~~
d) no, it must be a constant expression
~~e) yes, because x+25 has L-value~~

19. Does line 5 contain a valid case label?

- a) no, "A" is an address expression, not an int
b) yes, "A" has the value of the ASCII code of the letter 'A'
c) yes, any expression can be a case label

d) yes, "A" is a constant address, and all constants can be case labels

e) no, "A" is outside the range of x+25

20. Among lines 7, 9, 11, 13, 15, which lines contain valid case labels?

- ~~a) all lines~~ b) lines 7, 11 c) line 9 only
d) lines 7, 9, 15 ~~e) lines 7, 9, 11, 15~~

21. Which of the following is a correct truth table?

a	b	a b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a && b
0	0	0
0	1	0
1	0	1
1	1	1

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a && b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a b
0	0	1
0	1	1
1	0	1
1	1	0

22. Given the expression $\text{rand}() \% 10 - 5$, where rand() is the random number generator function you used for your assignment. What is the inclusive range of values for this expression?

- a) -5 to 4 b) 0 to 10 c) 5 to 14
d) 5 to 9 e) -5 to 5

Suppose you want an "accumulate" (累積) function, which keeps track of an accumulator value (initialized to 0), adds its integer parameter to the accumulator, and returns the new accumulator value. For example,

```

printf("%d ", accumulate(2));
printf("%d ", accumulate(5));
printf("%d ", accumulate(-4));
printf("%d ", accumulate(10));
printf("%d ", accumulate(8));

```

will print 2 7 3 13 21. Consider the following program

```

1 int accumulate(int n) {
2        a = 0; /* declare accumulator */
3   return     ;
4 }

```

23. What should be the content of the box on line 2?

- ~~b~~ a) extern int b) static int
c) ~~int~~ d) ~~auto int~~
e) global int

24. What should be the content of line 3?

- b. a) ++n b) a + n c) a ++ n
d) a = a + n e) a++

a. 25.

```
1 void Rec(int x) {  
2     if (x > 0) {  
3         Rec(x - 1);  
4     }  
5     printf("%d ", x);  
6 }
```

What is the output when you call `Rec(4)`?

- a) 0 1 2 3 4
b) 1 2 3 4
c) 4 3 2 1
d) 4 3 2 1 0
e) 4 3 2 1 0 1 2 3 4