University of Maryland College Park
Dept of Computer Science
CMSC106 Fall 2013
Midterm I Key

Last Name (PRINT): _____

First Name (PRINT): _____

University Directory ID (e.g., umcpturtle)_____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

Instructions

➢ This exam is a closed-book and closed-notes exam.
➢ Total point value is 100 points.
➢ The exam is a 50 minutes exam.
➢ Please use a pencil to complete the exam.
➢ WRITE NEATLY.
➢ You don't need to use meaningful variable names; however, we expect good indentation.

Grader Use Only

#1	Problem 1 (Miscellaneous)	(40)	
#2	Problem 2 (Conditionals)	(20)	
#3	Problem 3 (Loops)	(20)	
#4	Problem 4 (Loops)	(20)	
Total	Total (100)	(100)	

Problem #1 (40 pts)

1. (2 pts) Circle those identifiers considered valid in C?
   a. phone
   b. Lauren
   c. 17cat
   d. --age
   e. System###
   f. Sounds12

   Answer: a., b. f

2. (2 pts) Is the following an infinite loop? Briefly explain (yes or no answers will receive no credit).

   ```
   while(10 == 0) { printf("Here\n"); }
   ```

   Answer: No

3. (2 pts) The body of a *do while* will be executed:
   a. At least three times
   b. At least twice
   c. More than three times
   d. Never
   e. None of the above

   Answer: e.

4. (2 pts) In a for loop statement curly brackets are not required if we are executing:
   a. Only one statement
   b. Two statements
   c. Three statements
   d. None of the above

   Answer: a

5. (2 pts) How many values can you represent with 5 bits?

   Answer: 32

6. (2 pts) Write a Unix command that will copy the file data.in present in the directory **/system** to your home directory.

   Answer: cp /system/data.in ~

7. (2 pts) Write a Unix command that will compile the program my_test.c.

   Answer: gcc my_test.c

2

8. (2 pts) Which of the following prints the "%" symbol?
   a. printf("%%");
   b. printf("%");
   c. printf("\%");
   d. a. and b.
   e. a. and c.
   f. None of the above

   Answer: a.

9. (2 pts) Which of the following format specifiers skips spaces by default?
   a. %d
   b. %c
   c. %f
   d. a. and b.
   e. a. and c.
   f. None of the above

   Answer: e

10. (2 pts) Modify the following printf statement (not data) so we can generate the output: **7.276**

```
float data = 7.2758;
printf("%.3f", data);
```

   Answer: Everyone gets credit for this one.

11. (2 pts) Write the output generated by the following code fragment.

```
int x = 400, y = 3;
if (x == 200 || --y)
   printf("%d %d", x, y);
else
   printf("800\n");
```

   Answer: 400 2

12. (2 pts) Rewrite the following code using a single **if** statement.

```
if (day > 31)
   if (day <= 60)
      printf("February\n");
```

   Answer: if (day > 31 && day <= 60) printf ("February\n");

13. (8 pts) Rewrite the following code using a **switch** statement.

```
if (y == 10) {
   printf("Case 1\n");
} else if (y == 100) {
   printf("Case 2\n");
} else {
   printf("Unknown\n");
}
```

Answer:

```
switch(y) {
  case 10:
    printf("Case 1\n");
    break;
  case 100:
    printf("Case 2\n");
    break;
  default:
    printf("Unknown\n");
}
```

14. (2 pts) Write the output generated by the following code fragment if the value of x is 800.

```
if (x < 100)
   if (x < 80)
       printf("Dog\n");
else
   printf("Cat\n");
printf("Done");
```

Answer: Done

15. (2 pts) Rewrite the following statement using a single compound assignment operator.

```
y = y * 3;
y = y * 4;
```

Answer: y *= 12;

4

16. (4 pts) Rewrite the following code using a single **while** loop.

```
int x, y;
for (x = 1, y = 2; x < 10; x++, y++) {
   printf("%d %d\n", x, y);
}
```

Answer:

```
x = 1, y = 2;
while (x < 10) {
   printf("%d %d\n", x, y);
   x++;
   y++;
}
```

```
int x, y;
for (x = 1, y = 2; x < 10; x++, y++) {
   printf("%d %d\n", x, y);
}
```

Problem #2 (20 pts)

Write a complete C program that prints the route someone should take based on the hour of the day (hours will be between 1pm and 10pm). The program will read an **integer** value representing the hour and it will print a message according to the following rules:

"Alternate Route" → if the hour is between 1 and 7 (inclusive)
"Highway" → if the hour is after 7 and before 10 (inclusive)
"Any route" → any other value

For this program:
- You can use the message "Enter value" to read a value.
- You do not need to write pseudocode.

Answer:

```c
#include <stdio.h>

int main() {
   int hour;

   printf("Enter value: ");
   scanf("%d", &hour);

   if (hour >= 1 && hour <= 7) {
     printf("Alternate Route");
   } else if (hour > 7 && hour <= 10) {
     printf("Highway");
   } else {
     printf("Any route");
   }

   return 0;
}
```

Problem #3 (20 pts)

Write a C program that prints the powers of odd values between 1 and an integer value provided by the user. Use the message "Enter value: " to read the integer value.

The following is an example that illustrates the functionality of your program when the user enters 10. Notice that your program must work for other scenarios and not just the example below. For this problem you must use a while loop otherwise you will lose significant credit.

```
Enter value: 10
1    1
3    9
5    25
7    49
9    81
```

Answer:

```c
#include <stdio.h>

int main() {
   int value, i = 1;

   printf("Enter value: ");
   scanf("%d", &value);
   while(i <= value) {
      if (i % 2 != 0) {
         printf("%d     %d\n", i, i * i);
      }
      i++;
   }

   return 0;
}
```

Problem #4 (20 pts)

Write a program that reads a year value. The program expects a year value between 1 (inclusive) and 110 (inclusive). The program will generate the error message "Invalid year" and keep asking for a year value as long as the user provides an invalid one. Once a correct value has been provided, the program will print the number of days associated with the provided year value. Use the message "Total days: " followed by the number of days in order to display your answer.

The following is an example that illustrates the functionality of your program. Notice that your program must work for other scenarios and not just the example below. For this problem you must use a **do while** otherwise you will lose significant credit.

```
Enter years: -1
Invalid year
Enter years: 200
Invalid year
Enter years: 2
Total days: 730
```

Answer:

```c
#include <stdio.h>

int main() {
   int years, valid;

   do {
      printf("Enter years: ");
      scanf("%d", &years);
      valid = (years >= 1 && years <= 110);
      if (!valid) {
         printf("Invalid year\n");
      }
   } while(!valid);

   printf("Total days: %d\n", years * 365);

   return 0;
}
```