University of Maryland College Park
Dept of Computer Science
CMSC106 Fall 2016
Midterm I

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 200 points.
- The exam is a 50 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- **Your code must be efficient.**
- **You don't need to use meaningful variable names; however, we expect good indentation.**

Grader Use Only

#1	Problem #1 (Miscellaneous)	70	
#2	Problem #2 (Conditionals)	45	
#3	Problem #3 (Loops)	85	
Total	Total	200	

Problem #1 (Miscellaneous)

1. (3 pts) How many values can you represent with 4 bits?

2. (3 pts) Circle those identifiers considered VALID in C. We are not asking whether they are good identifiers names; we are asking whether they will compile.

   a. midterm#1
   b. highest_average
   c. 22_degrees
   d. Temperature#Control
   e. _processed

3. (3 pts) The body of a **do while** is executed at least:

   a. 2 times
   b. 1 time
   c. 3 or more times
   d. None of the above

4. (3 pts) Which of the following are considered false in C? Circle all those that apply.

   a. -14
   b. 1.56
   c. 0
   d. None of the above

5. (3 pts) Complete the following printf so it generates the output: **18.6342**

```
float y = 18.634237;

printf(                                          );
```

6. (3 pts) What is the output generated by the following program?

```
#include <stdio.h>

int main() {
   int y;

   printf("%d\n", y);

   return 0;
}
```

   a. 0
   b. A negative value.
   c. A positive value.
   d. A garbage/trash value (whatever is in the memory associated with y).
   e. None of the above.

7. (3 pts) Which of the following prints the "%" symbol? Circle all that apply.

   a. printf("%%");
   b. printf("%");
   c. printf("\%");
   d. None of the above

8. (3 pts) Rewrite the following statement using a single compound assignment operator.

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

9. (6 pts) Write a Unix command that will copy the file ice1.c present in the current directory to the 106 folder that is in your home directory.

10. (6 pts) Given the following code snippet:

```
int x, y = 20, m = 100;

scanf("%d", &x);

if (x > 0 || y--) {
   m++;
}
printf("%d %d\n", y, m);
```

a. What is the output if the user enters **-1**?

b. What is the output if the user enters **100**?

11. (6 pts) Complete the following printf statement so we can print the following message. Notice that the printf must print the quotes.

Robert "Bobby"

```
printf(                                    );
```

12. (12 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);
}
```

13. (16 pts) Rewrite the following code using a **switch** statement.

```
if (c == 'a') {
   printf("SC1\n");
} else if (c == 'm') {
   printf("SC2\n");
} else {
   printf("Other\n");
}
```

Problem #2 (Conditionals)

Write a C program that decides what kind of breakfast a person will have. First, the program will read an integer value that indicates whether the person is a vegetarian (number 1) or not (number 0). The program will read a second integer value representing the maximum number of calories a person can have. If the person is a vegetarian and the person can only have less than 100 calories, the breakfast choice will be fruit; otherwise the vegetarian will have oatmeal. If the person is not a vegetarian and can only have less than 200 calories, the breakfast choice will be a power bar; otherwise the person can have anything.

- Use the message "Enter 1 for vegetarian, 0 otherwise:" before reading the integer representing whether someone is a vegetarian or not.
- Use the message "Enter calories:" to read the calories value.
- The program must display as result "Fruit", "Oatmeal", "Power bar", or "Anything".
- You can assume the user will provide valid values.

Problem #3 (Loops)

Write a C program that implements a guessing game. A random value is obtained by calling a function named get_random_value() (see next page). The program will then read a value from the user and compare that value against the random value returned by the get_random_value() function. If the value provided by the user and the random value are the same, the player wins; otherwise the program will ask the user whether she/he would like to quit playing. If the user would like to quit, the program ends; otherwise the program will ask the user for another value. This process will continue until either the user quits or wins. For this program:

- You do not need to implement the get_random_value() function; you can assume someone has already implemented the function. This function returns an integer value.
- Notice that the get_random_value() function is only called once in order to get the value the user will guess. This value is compared against the values provided by the user.
- Use the message "Enter guess: " to read the user's guess.
- When the user provides the correct guess, the program will print the message "Correct guess: " followed by the number of guesses provided so far (including the correct guess). Notice that the program will end once the correct guess is provided.
- When the user provides an incorrect guess, the program will print the message "Wrong guess" and will ask whether the user would like to quit using the message "Do you want to quit? 0 for No, 1 for Yes:". If the user chooses to continue another value will be read; otherwise the program will end.
- **For this program you must use a do while, otherwise you will lose credit.**

Below we included examples of executing the program you are expected to implement. Remember, your program must work for other scenarios and not just the provide examples. Underlined text represents input provided by the user. The % represents the Unix prompt.

```
% a.out
Enter guess: 4
Wrong guess
Do you want to quit? 0 for No, 1 for Yes: 0
Enter guess: 8
Wrong guess
Do you want to quit? 0 for No, 1 for Yes: 1
% a.out
Enter guess: 4
Wrong guess
Do you want to quit? 0 for No, 1 for Yes: 0
Enter guess: 8
Wrong guess
Do you want to quit? 0 for No, 1 for Yes: 0
Enter guess: 10
Correct guess: 3 attempts
%
```

WRITE THE PROGRAM ON THE NEXT PAGE

```
#include <stdio.h>

int main() {
   int value_to_guess = get_random_value();
```

PAGE FOR YOUR ANSWERS

PAGE FOR YOUR ANSWERS