ECE 551D
Fall 2023
Test 1—Version 2

Name: NetID:

There are 5 questions, with the point values as shown below. You have 75 minutes with a total of 45 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. However, you are permitted one page of notes.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

#	**Question**	**Points**
1	Concepts	5
2	Reading Code	10
3	Testing	12
4	Algorithm	10
5	Writing Code	8
	Total	45

Question 1 Concepts [5 pts]

For all parts of this question, you *must* blacken the circle of the answer you choose. We can *only* see the region with the circles when grading.

1. Which of the following statements is *true* about an enumerated type in C?

   (a) A variable of enum type cannot be assigned a new value.

   (b) An enumerated type is most useful when you have a type of data with many values with conceptual names.

   (c) An enumerated type allows you to bundle multiple variables into a single entity.

   (d) Though enumerated types have integer values, they cannot be used in conditional statements.

2. Assuming a system with 8-bit integers, what value is assigned to `x` (in decimal) in the following expression?

```
unsigned x = 0x6A + 23
```

   (a) 128

   (b) -128

   (c) 129

   (d) -129

3. Which one of the following is *true* of object files?

   (a) They can be executed by a shell program.

   (b) They are the input of the linker in the compilation process.

   (c) They cannot have different names other than the default.

   (d) They are human-readable, containing the same information from the source files.

4. If you are following the scientific method for debugging and you are stuck trying to develop a hypothesis, which one of the following is the best way forward?

   (a) Gather more information.

   (b) Observe a phenomenon.

   (c) Test your hypothesis.

   (d) Ask a question.

5. Which one of the following is a benefit of using a build tool like make?

   (a) It detects memory errors in your program.

   (b) It has rules for each target that can save compiling time.

   (c) It helps you gather debugging information.

   (d) It helps you format your code uniformly.

Question 2 Reading Code [10 pts]

What is the output when the following C code is executed? (Assume appropriate header files have been included.)

```
1  struct _a_struct {
2    int a;
3    char c;
4  };
5  typedef struct _a_struct a_struct;

6  a_struct f(a_struct x, int z) {
7    if (z == 0) {
8      printf("Base case: x = {%d, %c}, z = %d\n", x.a, x.c, z);
9      return x;
10   }
11   x.a *= z;
12   x.c -= z;
13   if (z < 0) {
14     printf("z = %d (less than 0)\n", z);
15     z += 1;
16   }
17   else {
18     printf("z = %d (greater than 0)\n", z);
19     z -= 2;
20   }
21   printf("x = {%d, %c}, z = %d\n", x.a, x.c, z);
22   return f(x, z);
23 }

24 int main(void) {
25   a_struct x;
26   x.a = 10;
27   x.c = 'e';
28   double y = 3.6;
29   f(x, y);
30   printf("In main: x = {%d, %c}, y = %.1f\n", x.a, x.c, y);
31   return EXIT_SUCCESS;
32 }
```

Write your answer on the next page.

Your output should be 8 lines long. Please write each line where indicated below:

Output line 1
Output line 2
Output line 3
Output line 4
Output line 5
Output line 6
Output line 7
Output line 8

Question 3 Testing [12 pts]

Suppose the following code is correct to solve a certain problem:

```
1  int g(int x, int y, int z) {
2    int i;
3    if (x < 0) {
4      i = -x;
5    }
6    else if (x > y) {  // mistake  here: x < y
7      i = x + y;
8    }
9    else {
10     i = x;
11   }
12   while (z < 0) {
13     z += i;
14   }
15   return z;
16 }
```

Imagine that instead of this correct code, a programmer makes a mistake on line 6 and writes `x < y` instead of `x > y`.

Write *one* test case that would detect the above error (*i.e.*, exhibit different behavior than the correct program):

- Input:
- Expected return value:
- Return value if mistake were made:

Next, using the correct version of the code, write *two* additional test cases which, when combined with your case above, gives you **statement coverage** on the correct code:

x = y = z =

x = y = z =

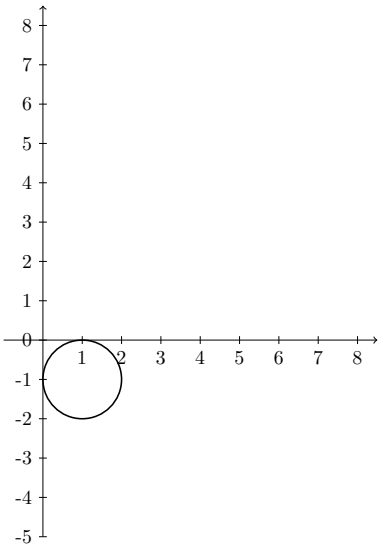
Do the three test cases you wrote above give **decision coverage** on the correct code? Why or why not?

Do the three test cases you wrote above give **path coverage** on the correct code? Why or why not?

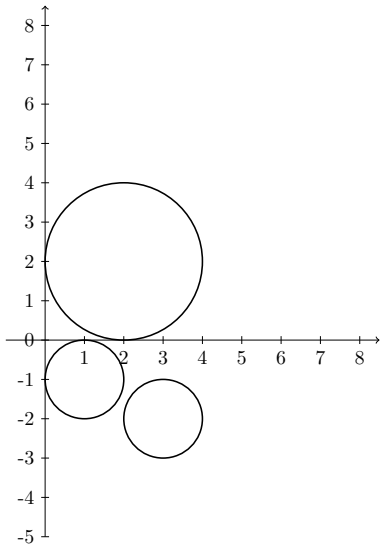
What kind of testing did you just do for this problem?

Question 4 Algorithm [10 pts]

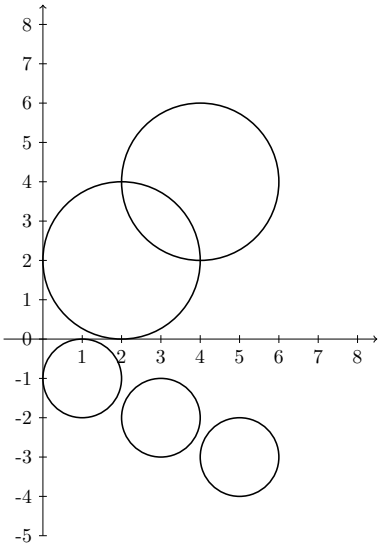
The figures below show the result of executing an algorithm with one parameter N, which must be a non-negative integer. For values of N from 0 to 3, the algorithm produces the following figures:



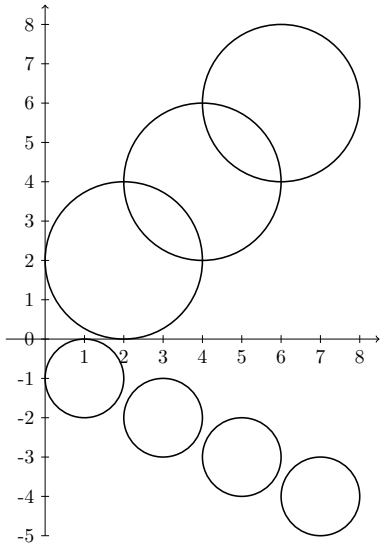
(a) $N = 0$



(b) $N = 1$



(c) $N = 2$



(d) $N = 3$

Fill in the blanks below to complete the algorithm used to generate these figures:

Given a non-negative integer N:

Count from _____ to _____ (exclusive), and

For each number that you count (call it i),

If _____ is even, then

Set y to be _____ .

Otherwise,

Set y to be _____ .

Draw a circle of radius _____ at point (_____ , y).

Question 5 Writing Code [8 pts]

Translate your algorithm from the previous question into C code. Make sure to specify the parameter(s). You may assume the function `draw_circle` is defined elsewhere:

```
void draw_circle(int x, int y, unsigned radius);

void draw_figure(                    ) {









}
```