Following is an example of the cover page you would see. The honor code instructions are for the real test, not the practice test. For the practice test, you may collaborate in any way, and there are no restrictions on resources. Please note that the real test will have more questions, and may have questions on different topics or in different formats.

ECE 551D
Fall 2021
Test 2—Version PRACTICE QUESTIONS

Name: | | NetID: | |

There are 4 questions, with the point values as shown below. You have 75 minutes with a total of 47 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. However, this exam is open notes, so you may use your class notes, which must be handwritten by you.

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	Reading Code (Fall 2016 Midterm)	9
2	Coding Pictionary (Fall 2013 Final)	12
3	Coding with Arrays (Fall 2016 Midterm)	8
4	Coding (Fall 2017 Midterm)	18
	Total	47

Question 2 Reading Code [9 pts]

Execute the following code by hand, and fill in the output at the bottom of this page (the start of each line is written for you, write the correct number at the end of each line).

```
#include <stdio.h>
#include <stdlib.h>
void f(int x, int * p, int ** q) {
  x = x + 3;
  *p = **q + x;
  *q = p;
}
int main(void) {
  int a = 3;
  int b = 4;
  int c = 5;
  int * data[] = {&a, &b, &c};
  int ** q = &data[1];
  **q = 99;
  q[0] = q[1];
  printf("a=%d, b=%d, c=%d\n",a,b,c);
  for (int i = 0; i < 3; i++) {
    *data[i] = *data[i] + 11;
  }
  printf("a=%d, b=%d, c=%d\n",a,b,c);
  f(a, &b, &data[2]);
  *q[1] = *q[1]-3;
  printf("a=%d, b=%d, c=%d\n",a,b,c);
  return EXIT_SUCCESS;
}
```

Fill in output below:

a= , b= , c=

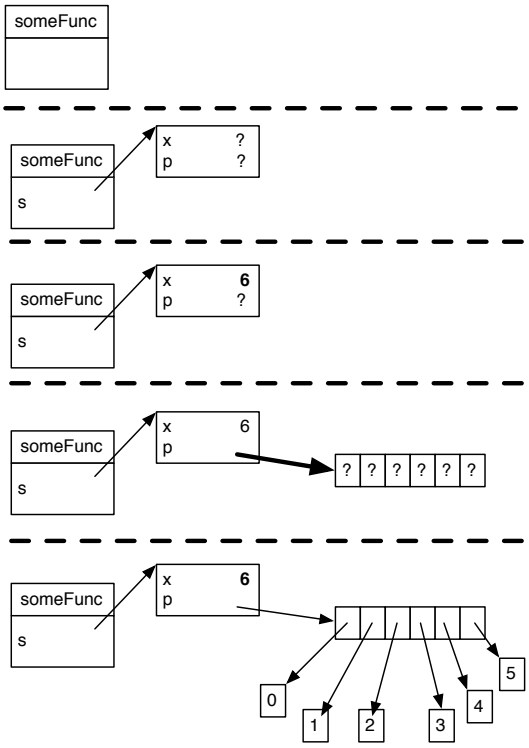
a= , b= , c=

a= , b= , c=

Question 5 Coding Pictionary [12 pts]

The figure below on the right depicts the state of `someFunc` at five points in time. Each dotted horizontal line separates one time from the next. You should write `someFunc` (on the left) by writing the code that goes with the four dotted lines—that is, your `someFunc`'s execution should look like the pictures on the right:

```
struct _astruct {
   int x;
   int ** p;
};
typedef struct _astruct astruct;

astruct * someFunc(void) {
//code for the first dotted line



//code for the second dotted line



//code for the third dotted line



//code for the fourth dotted line



 return s;
}
```



Question 5 Coding 2: Arrays [8 pts]

Write the method `findLargestDifference` which takes two arrays of integers, `a1` and `a2`, and `n` which is the size of both arrays. This method returns the largest difference (not the largest absolute value of the difference, the largest numerical difference) between elements in the same positions in `a1` and `a2`. For example, if

```
a1 = {1, 4, 6, 7, 9}
a2 = {7, 5, 1, 3, 8}
```

Then this method would return 5, since the differences between elements in the same positions are -6, -1, 5, 4, and 1, respectively, and 5 is the largest of those differences. If `n` is zero, your method should return 0.

```
int findLargestDifference(int * a1, int * a2, size_t n) {




}
```

Question 7 Coding 4: A Program [18 pts]

For this problem, you are going to write a program which takes one command line argument—the name of an input file to read—reads all the lines in it, and prints each unique line. If an identical line appears twice in the file, your program should print only the first occurrence. For example, given

```
apple
banana
apple
cat
Banana
b a n a n a
banana
```

Your program should print

```
apple
banana
cat
Banana
b a n a n a
```

Note that this program only consider strings the same if they are **exactly** the same (differences in case and spacing matter). For this problem, you may assume:

- The correct number of command line arguments are given
- `fopen`, `fclose`, `malloc`, and `realloc` always succeed.

You should write your answer on the next page

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//YOUR CODE GOES HERE
```