ENEE 140, Fall 2022
Midterm Exam — Answer Key
Do Not Make a Copy!!

Date: Tuesday, October 25, 2022, 9:30 – 10:45 am

University of Maryland Honor Pledge: The University is committed to Academic Integrity, and has a nationally recognized Honor Code, administered by the Student Honor Council. In an effort to affirm a community of trust, the Student Honor Council proposed and the University Senate approved an Honor Pledge. The University of Maryland Honor Pledge Reads:

"*I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment)*"

Please write the exact wording of the Pledge, followed by your signature, in the space below:

Pledge: _____
Pledge: _____
Pledge: _____
Pledge: _____

Your signature: _____
Full name: _____ Course: _____ Directory ID: _____

List of Exam Questions:

Question:	1	2	3	4	5	6	Total
Points:	24	4	18	18	15	21	100
Score:							

Instructions:

- Make sure that your exam is not missing any sheets, then write your full name, your section and your Directory ID on the front.
- Write your name and section at the bottom of each page as well.

- Write your answers in the space provided below the problem. If you make a mess, clearly indicate your final answer.
- The exam has a maximum score of 100 points.
- The problems are of varying difficulty. The point value of each problem is indicated. Pile up the easy points quickly and then come back to the harder problems.
- This exam is OPEN BOOK. You may use any books or notes you like. Calculators are allowed, but no other electronic devices. Good luck!

1. (24 points) This problem tests your understanding of C types and casts and of C operators. Assume that variables `a`, `b`, `c` and `d` are defined as follows:

```
int      a = 1;
char     b = '2';
float    c = 3;
unsigned d = 4;
```

Fill in all the empty cells in the table below. For each of the C assignment expressions in the left column, state the resulting value of the `r1`–`r8` variables. If an expression results in a compilation error, write ERROR.

Assignment		Value
int	r0 = d / 2;	2
int	r1 = d / 3;	**1**
float	r2 = d / 3;	**1.0**
int	r3 = c / 2 + 0.6;	**2**
int	r4 = b − '0';	**2**
unsigned	r5 = d / a++;	**4**
unsigned	r6 = c % d;	**ERROR**
unsigned	r7 = UINT_MAX + d;	**3**
int	r8 = INT_MAX * 2;	**-2**

2. (4 points) This problem tests your understanding of **for** loops. Choose the correct **for** loop for iterating over each even index of an array of length 14 (Assume 0 is even and that `i` is an integer variable, correctly declared).

A. **for** (i = 1; i < 14; i+=2)

B. **for** (i = 0; i <= 13; i++)

C. **for** (i = 0; i < 14; i+2)

D. **for** (i = 0; i < 14; i+=2)

2. **D**

Solution: **for** (**int** i = 0; i < 14; i+=2)

Because the array is of length 14, the highest index is 13.
This means we need to stop iterating once 14 is reached.
To only access even indices, start at 0 and increment by 2.

3. (18 points) This question tests your understanding of loops and data input. It has two parts.

   (a) Fill in the blanks so that the following program prints the squares of each number from 0 to an integer read using `scanf`. The square of a number is equal to the number multiplied by itself.

```
#include <stdio.h>
int main(){
    int iter, read;
    scanf("%____d____", ____&read____);
    for(iter____=0____; iter____<=read____; iter____++____){
        printf("The square of %d is: ", ____iter____);
        printf("%d\n", ____iter*iter____);
    }
    return 0;
}
```

Solution:

```
#include <stdio.h>
int main(){
    int iter, read;
    scanf("%d", &read);
    for(iter= 0; iter <= read; iter++){
        printf("The square of %d is: ", iter);
        printf("%d\n", iter*iter);
    }
}
```

   (b) Rewrite the **for** loop as a **while** loop.

Solution:

```
iter = 0;
while(iter <= read){
        printf("The square of %d is: ", iter );
        printf("%d\n", iter*iter);
        iter ++;
}
```

4. (18 points) This question tests your ability to debug code. The following program should find the average of an array of 5 floats. It should then print this average out to 1 decimal place of accuracy. Find the bugs in the code. Hint: There are bugs on 5 lines.

```
1  #include <stdio.h>
2
3  int main () {
4    int i;
5    float array[5] = [1.2, 3.5, 4.2, 9.8, 5.0];
6    int avg, sum;
7
8    for (i=1;i <= 5; i++) {
9      sum += array[i];
10   }
11
12   avg = sum / 5
13
14   printf("%1d\n", &avg);
15   return 0;
16 }
```

Solution:

```
Line 5: float array[5] = {1.2, 3.5, 4.2, 9.8, 5.0};
        Array values are defined with {} not [].
Line 6: float avg, sum;
        avg and sum should be floats to calculate an accurate average.
Line 8: for (i=0;i < 5; i++) { or for (i=0;i <= 4; i++) {
        The current loop would iterate past the end of the array.
Line 12: avg = sum / 5;
        Missing semicolon.
Line 14: printf("%.1f\n", avg);
        Incorrect format specifier. Use of & is incorrect.
```

5. (15 points) This question tests your understanding of C variables and operations. In the program below, the variable `FLUBBER` will be having the ride of its life. Write down the output of the program.

   To do this, you should track the progress of `FLUBBER` across the program, keep in mind all changes to its value, and write down what what each of the `printf` statements will print out.

```
1  #include <stdio.h>
2
3  void mystery_function(int FLUBBER);
4
5  int main() {
6      int FLUBBER = 0;
7      int i;
8
9      for (i = 0; i < 6; i++) {
10         FLUBBER += i;
11     }
12     printf("FLUBBER = %d\n", FLUBBER);
13
14     mystery_function(FLUBBER);
15     printf("FLUBBER = %d\n", FLUBBER);
16
17     FLUBBER += (10.0 / 3.0);
18     printf("FLUBBER = %d\n", FLUBBER);
19
20     FLUBBER += 0x31;
21     printf("FLUBBER = %d\n", FLUBBER);
22
23     FLUBBER += 2*(FLUBBER & 1);
24     printf("FLUBBER = %d\n", FLUBBER);
25
26     return 0;
27 }
28
29 void mystery_function(int x) {
30     int i = 0;
31     while (i < 2) {
32         x = (x*2) - i;
33         i += 1;
34     }
35     if (x % 2 == 0) {
36         x += 10;
37     } else {
38         x += 5;
39     }
40 }
```

Solution:

```
LN12: FLUBBER = 15
LN15: FLUBBER = 15 (mystery_function does not change FLUBBER in main())
LN18: FLUBBER = 18 (should be +3)
LN21: FLUBBER = 67 (should be +49)
LN24: FLUBBER = 69 (should be +2)
```

6. (21 points) This question will test your understanding of strings. The intent of the program below is to read in a user's first name, capitalize the first character in the name, and print it out. It is expected that the user will input no more than 10 characters for their first name. You can assume that the user will enter only alphabetical and lowercase characters. Fill in the blanks to complete the program.

   Ex. For an input of "john", "John" will be printed out.

```
#include <stdio.h>

int main() {
    char first_name[      11      ];

    printf("Enter your first name: ");
    scanf("%      s      ",  first_name  );


    first_name[      0      ] = first_name[      0      ] +  ('A'-'a')  ;
    printf("Your first name is: %      s      \n", first_name);

    return 0;
}
```

Solution:

```
#include <stdio.h>

int main() {
```

```
    char first_name[11];

    printf("Enter your first name: ");
    scanf("%s", first_name);

    first_name[0] = first_name[0] + ('A'-'a');
    printf("Your first name is: %s\n", first_name);

    return 0;
}
```