

ENEE 140, Spring 2016  
 Final Exam — Answer Key  
**Do Not Make a Copy!!**

**Date:** Wednesday, May 18, 10:30 am – 12:30 pm

**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	7	8	9	Total
Points:	16	6	6	10	20	10	10	12	10	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 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!

- (16 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 = 4;
unsigned b = 3;
float    c = 2;
float    d = 1;

```

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 `r2-r9` variables. If an expression results in an error, write ERROR.

Assignment		Value
<b>int</b>	<code>r1 = a/b;</code>	1
<b>float</b>	<code>r2 = a/b;</code>	1
<b>int</b>	<code>r3 = b/c;</code>	1
<b>int</b>	<code>r4 = (b-a) &gt; 0;</code>	1
<b>float</b>	<code>r5 = a*d/8;</code>	0.5
<b>int</b>	<code>r6 = a&gt;&gt;1;</code>	2
<b>int</b>	<code>r7 = INT_MIN % 2;</code>	0
<b>unsigned</b>	<code>r8 = b % a;</code>	3
<b>unsigned</b>	<code>r9 = (unsigned)(d-c) % UINT_MAX;</code>	0

- (6 points) This problem tests your understanding of C operations. What is the output of the following program?

```

#include <stdio.h>

int main() {
    int x = 7;
    int y = x++ + 3;
    int z = 7 * --x + 4;
    printf("x=%d\ny=%d\nz=%d", x, y, z);
}

```

**Solution:**

```

x = 7
y = 10
z = 53

```

- (6 points) This problem tests your understanding of random number generation. The following program generates 5 random numbers:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    int i = 0;
    int r;

```

```

    srand(time(NULL));
    for (i = 0; i < 5; i++)
        r = _____;
    return 0;
}

```

How should you fill the blank so that the program generates:

(a) A number from 0 to 99

(a) \_\_\_\_\_ **r = rand() % 100** \_\_\_\_\_

(b) A number from -49 to 0

(b) \_\_\_\_\_ **r = (-1\*(rand() % 50))** \_\_\_\_\_

(c) An odd number from 80 to 100

(c) \_\_\_\_\_ **r = (2\*(rand() % 10))+81** \_\_\_\_\_

4. (10 points) This problem tests your understanding of strings. The following program should reverse a string with no spaces. Fill in the blanks so that the program works.

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

```

int main()
{
    char original[100], reverse[100];
    int size, c, d;

    printf("Input a string\n");
    scanf(-----, -----); // puts the input into the original array

    //This for loop finds the size of the string. It may not be 100!
    for (c = 0; -----; -----) {
        if (original[c] == -----) {
            size = c;
            break;
        }
    }

    d = 0;
    // This for loop puts the elements of original into reverse
    for (c = -----; -----; c--) {
        ----- = -----
        d++;
    }

    reverse[d] = -----;

    printf("%s\n", reverse);
}

```

```

    return 0;
}

```

**Solution:**

```

#include <stdio.h>

int main()
{
    char original[100], reverse[100];
    int size, c, d;

    printf("Input a string\n");
    scanf("%s", original);

    for (c = 0; c < 100; c++) {
        if (original[c] == '\0') {
            size = c;
            break;
        }
    }
    d = 0;
    for (c = size - 1; c >= 0; c--) {
        reverse[d] = original[c]; // OR reverse[c] = original[d];
        d++;
    }

    reverse[d] = '\0';

    printf("%s\n", reverse);

    return 0;
}

```

5. (20 points) This problem tests your understanding of C arrays. Fill in the code to store the following triangle into a two dimensional array and then print it.

```

*
***
*****
*****
*****

```

```

#include <stdio.h>

```

```

int main()
{
    char shape[5][9];

```

```

int r, c, i, mid;

mid = 4; // 4 is the median of 9 indices

for (r = 0; -----; r++)
{
    for (c = 0; -----; c++) {
        shape[r][c] = ' ';
    }
    for (i = 0; -----; i++) {
        shape[r][c] = '*';
        c++;
    }
    for (----- c < 9; -----) {
        shape[r][c] = ' ';
    }
    mid = -----
}

for (-----) { // This for loop prints out the triangle from the array
    for (-----)
        printf(-----);
    printf(-----);
}

    return 0;
}

```

**Solution:**

```

#include <stdio.h>

int main()
{
    char shape[5][9];

    int r, c, i, mid;

    mid = 4;

    for (r = 0; r < 5; r++)
    {
        for (c = 0; c < mid; c++) {
            shape[r][c] = ' ';
        }
        for (i = 0; i < 2*r+1; i++) {
            shape[r][c] = '*';
            c++;
        }
        for (; c < 9; c++) {

```

```
        shape[r][c] = ' ';
    }
    mid--;
}

for (r = 0; r < 5; r++) {
    for (c = 0; c < 9; c++)
        printf("%c", shape[r][c]);
    printf("\n");
}

return 0;
}
```

6. (10 points) This question tests our understanding of 2D arrays. Determine the output of the following program.

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

void print_array(int array[3][3]);

int main(void)
{
    int array[3][3] = {0};
    int i, j;

    for(i=0; i<3; i++){
        for(j=0; j<i; j++){
            array[i][j]++;
        }
    }
    print_array(array);

    for(i=0; i<3; i++){
        for(j=i; j<3; j++){
            array[j][j]++;
        }
    }
    print_array(array);

    for(i=1; i<3; i++){
        for(j=0; j<3; j++){
            array[i][j] = array[i-1][j];
        }
    }
    print_array(array);
}
```

```
        return 0;
    }

    void print_array(int array[3][3])
    {
        int i, j;
        for(i=0; i<3; i++){
            for(j=0; j<3; j++){
                printf("%d_", array[i][j]);
            }
            printf("\n");
        }
        printf("\n");
    }
}
```

Indicate the program output below; consider that each line corresponds to a line of text on the console.

**Solution:**

```
0 0 0
1 0 0
1 1 0

1 0 0
1 2 0
1 1 3

1 0 0
1 0 0
1 0 0
```

7. (10 points) This question tests your understanding of file input/output. Give a high-level description of what the program does. Do not describe the program line-by-line.

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

```
int main (int argc, char *argv[])
{
    FILE *in;
    int input_c;
    int a = 0, b = 0;

    if(argc == 2){
        if( (in = fopen(argv[1], "r")) == 0 ) {
            printf("fopen_failed\n");
            return 0;
        }
    }
}
```

```
    else {
        printf("incorrect arguments\n");
        return 0;
    }

    input_c = fgetc(in);
    do{
        if(input_c != '_' && input_c != '\n'){
            a++;
        }
        else{
            if(a > b){
                b = a;
            }
            a = 0;
        }

        input_c = fgetc(in);
    } while( input_c != EOF);
    if(a > b){
        b = a;
    }

    printf("b=_%d\n", b);

}
```

Write your answer in the box below.

**Solution:**

```
// answer: opens a file , determines the length of its longest word
```

8. (12 points) This problem tests your understanding of control flow and variable scope. What does the following program print out?

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

```
int a;
```

```
void
```

```
f1(int a)
```

```
{
```

```
    a = a % 2 ? a + 2 : a;
```

```
    printf("%c", 'a' + a);
```

```
}
```

```
void
```

```
f2()
```

```
{
```

```
    static int a = 0;
```



```

    f1(a++);
}

int
main(int argc, char *argv[])
{
    int i;
    a = 0;

    for (i=0; i<2; i++) {
        f1(a);
        f2();
        a++;
    }

    return 0;
}

```

**Solution:**

Correct answer:

aadd

Possible incorrect answers (and the corresponding mistakes):

aadf (a from f1 is the global a)

aaca (a from f2 is the global a)

aded (a from both functions are the global a)

aada (a from f2 is local but is reset at each invocation)

aacc (a from f2 is the global a and is not reset at each invocation)

aabb (the ?: operator has no effect)

adeh (a from f2 is the global a, it is not reset at each invocation, and a from f1 is the global a)

aacc (a from f2 is the global a, it is not reset at each invocation, and the ?: operator has no effect)

9. (10 points) This problem tests your understanding of sorting. The function listed below is supposed to sort the first  $n$  elements of the array  $a$  in a descending order ( $a[0] \geq a[1] \geq a[2] \geq \dots \geq a[n-1]$ ) using the Selection Sort algorithm. Find all the bugs in the implementation; for each bug, indicate the line number where it occurs.

```

1 void
2 selection_sort(int n, int a)
3 {
4     int i, j;
5
6     for (i=n-1; i>=0; i--) {
7         for (j=i; j<n; j++) {
8             if (a[i] > a[j]) {
9                 tmp = a[i];
10                a[j] = tmp;
11                a[i] = a[j];

```

```
12     }
13     }
14 }
15
16     return 0;
17 }
```

**Solution:**

```
void
selection_sort(int n, int a[]) // a must be an array
{
    int i, j, tmp;           // tmp must be declared

    for (i=n-1; i>=0; i--) {
        for (j=0; j<i; j++) { // the unsorted part of a is between 0 and i
            if (a[i] > a[j]) {
                tmp = a[i];
                a[i] = a[j]; // assignments interchanged in variable swap
                a[j] = tmp;
            }
        }
    }
} // the function does not return a value
```