ENEE 140, Fall 2018
Midterm Exam — Answer Key
Do Not Make a Copy!!

Date:

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	Total
Points:	16	21	8	10	10	15	20	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!

1. (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:

```
char     a = '0';
int      b = 1;
unsigned c = 2;
float    d = 3;
```

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 a compilation error, write ERROR.

Assignment		Value
float	r0 = d / 2;	1.5
float	r1 = d / c;	**1.5**
float	r2 = b / c;	**0**
int	r3 = d / (b+c);	**1**
int	r4 = c++ % 2;	**0**
int	r5 = d % 2;	**ERROR**
char	r6 = a + b;	**'1'**
unsigned	r7 = UINT_MAX + 3;	**2**
int	r8 = b >> 1;	**0**

2. (21 points) This problem tests your understanding of `for` loops. Fill in the blanks to finish the program. Its output should be a right triangle made out of asterisks with sides of 8:

```
********
 *******
  ******
   *****
    ****
     ***
      **
       *
```

```
#include <stdio.h>

int
main()
{
  int i, j;
  for (   i = 0   ;   i < 8   ; i++){
    for(   j = 0   ;   j < 8   ; j++) {
      if (   j < i   )
        printf(   " "   );
      else
        printf(   "*"   );
```

```
    }
    printf("\n");
  }
  return 0;
}
```

Solution: Alternative implementation

```
#include <stdio.h>

int
main()
{
  int i, j;
  for (i = 0]; i < 8; i++){
    for(j = 0; j < 8; j++) {
      if (j >= i)
        printf("*");
      else
        printf("␣");
    }
    printf("\n");
  }
  return 0;
}
```

3. (8 points) This problem tests understanding of functions. Given the following code, write the appropriate function prototype for `make_uid()`:

```
int main () {
      float student_height;
      int student_age;
      char first_initial;
      unsigned student_UID;

      // Function that determines a student's UID based on height, age,
      // and their first initial
      student_UID = make_uid(first_initial, student_height, student_age);
      return 0;
}
```

Solution:

```
unsigned make_uid(char first_initial, float student_height, int student_age);
```

4. (10 points) This problem tests your understanding of characters and of number representation. Consider the function `decToHex()`, which takes in a decimal number and prints its hexadecimal (base 16) representation, with the digits in reverse order.

```
void decToHex(unsigned decimal) {
  int remainder;
  char digit;
  while ( decimal > 0 ) {
    remainder = decimal % 16;
    decimal = decimal / 16;
    if ( remainder < 10 ) {
      digit = '0' + remainder;
    }
    else {
      digit = 'A' + (remainder-10);
    }
    printf("%c", digit);
  }
  return;
}
```

For example, if you call `decToHex(140)`, the function should print `C8`, because the hexadecimal representation of 140 is `0x8C`.

Fill in the blanks with the correct comparison operators to ensure that the output of the function is correct.

5. (10 points) This problem tests your understanding of program input/output and of computer arithmetic. Suppose the user compiles the following program, runs it, and inputs "9 5" into the command line. What will be the output of the program? (indicate a space with an underscore).

```
#include <stdio.h>

int main() {
  int a, b;
  float c, d;

  scanf("%d %d", &a, &b);

  c = a/b;
  d = 1.0*a/b;

  printf("%5.2f %.2f\n", c, d);
}
```

Solution:

```
_1.00_1.80
```

6. (15 points) This question tests your understanding of for-loop execution and if-statements. What is the output after this program is executed?

```
#include "stdlib.h"
#include "stdio.h"

int foo1(int x);

int main(){


  int i;
  for(i=0;i<6;i++){
    if(foo1(i)%2==0){
      printf("%d\n",i);
    }
        i=i*2;

  }

  return 0;
}

int foo1(int x){

  x=x/2;
  printf("Entered function\n");
  return x;
}
```

Solution:

```
Entered function
0
Entered function
1
Entered function
```

7. (20 points) This problem tests your understanding of loops and program variables. Given the following code, please write the output:

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

```
int main(){
        int i=20;
        int j=0;
        int x = i*j;


        while((i*j++)<100){
                if(j%2==0)
                x=i*j;
                printf("j = %d\tx = %d\n",j,x);
        }
        return 0;
}
```

Solution:

```
j = 1    x = 0
j = 2    x = 40
j = 3    x = 40
j = 4    x = 80
j = 5    x = 80
```