ENEE 140, Fall 2025
Final 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	8	9	10	11	12	Total
Points:	16	5	8	8	10	3	4	20	8	8	10	12	112
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 last question is for EXTRA CREDIT and is worth 12 points. You may receive full credit without answering it, assuming that you answered correctly all the other questions (the exam will be scored out 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. No electronic devices (e.g. laptops, tablets, smartphones, calculators) are allowed. Good luck!

1. (16 points) This problem tests your understanding of C types, casts, and C operators. Assume that variables a, b, c, and d are defined as follows:

```
int          a = 1;
unsigned     b = 256;
char         c = 'f';
float        d = 2.72;
```

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
float	`r0 = d + 2;`	4.72
int	`r1 = b >> 3;`	**32**
char	`r2 = c + 1;`	**'g'**
float	`r3 = d % a;`	**ERROR**
unsigned	`r4 = b << 1;`	**512**
int	`r5 = a ^ b;`	**257**
float	`r6 = d - (int) d;`	**0.72**
unsigned	`r7 = (a + b) & 10;`	**0**
int	`r8 = (int)d >> 1`	**1**

2. (5 points) This question test your understanding of various concepts in the class. Fill in the blanks with the most appropriate answer.
   (a) Using bitwise AND with some number and 1 returns the value of the **least** significant bit of that number.
   (b) **continue** returns the program to the top of the loop containing it.
   (c) The result of dividing two integers is a/an **integer**.
   (d) A **do-while** loop will always run at least once.
   (e) **Function prototypes** define the return type and parameters of a function without implementation.

3. (8 points) This question tests your understanding of hash tables from Project 2.

   Consider a hash table of size **5** using **closed hashing with linear probing** to handle collisions (e xactly like in the project). You will insert the following string keys in order:

   "apple", "mango", "berry", "pear"

   The initial hash indices for these keys are:

Key	Initial Hash Index
apple	2
mango	2
berry	4
pear	2

   Using linear probing (with wrap-around), fill in the final hash table. Enter the key names (e.g. apple, mango, etc) at their correct final index in the hash table. Use "–" for empty cells.

Index	**0**	**1**	**2**	**3**	**4**
Key					

Solution:

Index	0	1	2	3	4
Key	pear	−−	apple	mango	berry

4. (8 points) This is a four-part multiple choice. These questions test various concepts of C. Some questions may have more than one correct answer.

   (a) Which keyword is used to create a structure?

   A. structs

   B. struct

   C. str

   D. structure

   (a) **B**

   Solution: B

   (b) Which of the following is not a logical operator?

   A. `&&`

   B. `|`

   C. `||`

   D. !

   (b) **B**

   Solution: B

   (c) How is an array initialized in C?

   A. int a[3] = {1,2,3};

   B. int a = 1,2,3;

   C. int a[] = new int[3];

   D. int a(3) = [1,2,3];

   (c) **A**

   Solution: A

   (d) According to Tudor, a good C program is a combination of what components?

   A. Functions

   B. Variables

C. Statements

D. Comments

(d) **A,B,C,D**

Solution: A,B,C,D

5. (10 points) This C program modifies the string "Hello World" character by character. Trace the program's execution and write down its output.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char str[] = "Hello World";
6
7      for(int i = 0; str[i] != '\0'; i++) {
8          if(str[i] >= 'A' && str[i] <= 'Z') {
9              str[i] = ((str[i] - 'A' + 3) % ('Z' - 'A' + 1)) + 'A';
10         } else if(str[i] >= 'a' && str[i] <= 'z') {
11             str[i] = ((str[i] - 'a' + 3) % ('z' - 'a' + 1)) + 'a';
12         }
13     }
14
15     printf("Encrypted name: %s\n", str);
16     return 0;
17 }
```

Solution:

```
Encrypted name: Khoor Zruog
```

6. (3 points) The following code segment tests your understanding of structs and printf formatting. Fill in the blanks to print the specified output.

```
typedef struct student {
        char name[1000];
        char grade;
        float score;
    } student;
    struct student s1;

    printf("Name is: %_____s_____\n", ____s1.name____);
    printf("Grade is: %_____c_____\n", ____s1.grade____);
    printf("Score is: %_____f_____\n", ____s1.score____);;
```

Solution:

```
printf("Name is: %s\n", s1.name);
printf("Grade is: %c\n", s1.grade);
```

```
    printf("Score is: %f\n", s1.score);
```

7. (4 points) This question tests your understanding of command line arguments. Assume that you are provided a file called **secret.c** with the following content:

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

int main(int argc, char *argv[]) {

    if (argc < 3) {
        printf("Not enough arguments provided!\n");
        exit(0);
    }
    else if (argc == 3) {
        printf("The name is %s. %s %s\n", argv[2], argv[1], argv[2]);
    }
    return 0;
}
```

Suppose secret.c was compiled as follows: gcc -o secret secret.c
What is printed when the following commands are ran?

(a) `./secret Detective Sherlock Holmes`

A. The name is Detective. Sherlock Holmes

B. Not enough arguments provided!

C. The name is Sherlock. Detective Sherlock

D. (No output)

(a) D

Solution: D

(b) `./secret James Bond`

A. Not enough arguments provided!

B. The name is Bond. James Bond

C. The name is James. James Bond

D. (No output)

(b) B

Solution: B

8. (20 points) This question tests your understanding of command line arguments, file I/O, and structures. The following text file of the present catalog, `present_cat.txt`, is provided below:

```
Choice #1:
Box of Chocolates
Price (in Dollars):
10.99
Choice #2:
LEGO Set
Price (in Dollars):
49.99
Choice #3:
New Computer
Price (in Dollars):
549.99
```

Fill in the blanks in the following `present_catalog.c` code so that the file is read in and each Choice is stored in their own gift structure.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  #define MAX_LINE 375
5
6  typedef struct {
7      int choice;
8      char name[MAX_LINE];
9      float price;
10 } Gift_st;
11
12 int main(int argc, char ____argv____[][])
13 {
14     // Data Structure Declared
15     Gift_st gift[_____3_____];
16
17
18     // Placeholder String Declared
19     char line[MAX_LINE];
20     // Placeholder Number
21     int temp = 0;
22     // Placeholder Floating Point
23     float value = 0;
24
25     if(____argc____ != 2) {
26         printf("Usage: ./present_catalog <file>\n");
27     }
28
29
30     FILE* present_cat;
```

```
31      present_cat = fopen(    argv[1]    , "r");
32      if (  present_cat   == NULL) {
33          printf("Present Catalog not found\n");
34          return 1;
35      }
36      int i;
37      for(i = 0; i < 3; i++) {
38          fgets(line, MAX_LINE, present_cat);
39          sscanf(line, "Choice #%d", &temp);
40          gift[i].choice  =     temp     ;
41
42          fgets(line, MAX_LINE, present_cat);
43          strcpy( gift[i].name , line);
44
45          fgets(line, MAX_LINE, present_cat);
46          fgets(line, MAX_LINE, present_cat);
47          sscanf(line, "%f", &value);
48          gift[i].price =     value     ;
49      }
50      fclose(present_cat)
51      return 0;
52  }
```

Solution:

Line 12: argv
Line 15: 3
Line 25: argc
Line 31: argv[1]
Line 32: present_cat
Line 40: gift[i].choice , temp
Line 43: gift[i].name
Line 48: gift[i].price , value

9. (8 points) This question tests your understanding of switch statements. Convert the sequence of if/else statements to a single switch statement.

```
if (i == 0 || i == 1) {
    printf("Hello!");
}

else if (i == 4) {
    printf("Hi!");
}

else if (i == 5) {
    printf("Hey!");
}

else if (i == 8 || i == 9 || i == 10) {
    printf("Howdy!");
}

else {
    printf("Bye!");
}
```

Solution:

```
switch (i) {
    case 0:
    case 1: printf("Hello!"); break;
    case 4: printf("Hi!"); break;
    case 5: printf("Hey!"); break;
    case 8:
    case 9:
    case 10: printf("Howdy!"); break;
    default: printf("Bye!"); break;
}
```

10. (8 points) Debugging: This problem will test your understanding of loops and arrays. There are 5 bugs on 5 lines, including the below:

    Line 1: #Include should be #include

    The intent of this program is to enter any number and store the following three even numbers in an array of length 4.

    For example, if the user enters "4" the program should store 6, 8, and 10 in the array. Find all errors and make an appropriate correction. The errors can be syntax or logical. There are no bugs on line 14.

```
Enter a number: 4
arr[1] is 6
arr[2] is 8
arr[3] is 10
```

```
1  #Include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main(){
6
7      int i;
8      int arr[];
9
10     printf("Enter a number: ");
11
12     scanf("%c", &arr[0]);
13
14     for (i = 1; i <  4; i++){
15         arr[i] = arr[0] + (i - 1)*2;
16         printf("arr[%d] is %d\n",i,&arr[i]);
17     }
18
19     return 0;
20 }
```

Solution:

Line 8: arr[] should be arr[4]
Line 12: "%c" should be "%d"
Line 15: (i-1)*2 should be (i)*2
Line 16, &arr[i] should be arr[i]

```
//Corrected code:
#include <stdio.h>
#include <stdlib.h>
```

```
int main(){

    int a, i;
    int arr[4];

    printf("Enter a number: ");


    scanf("%d",&arr[0]);


    for (i = 1; i <  4; i++){
        arr[i] = arr[0] + (i)*2;
        printf("arr[%d] is %d\n",i,arr[i]);
    }
    return 0;
}
```

11. (10 points) This question tests your understanding of loops, arrays, and aggregates. What is the output of the program?

```c
#include <stdio.h>

int main(void) {
    int a[] = {42, 6, 44, 51, 6, 25, 92, 33, 81, 100};
    int i, gt_num = 0;

    for (i = 0; i < 10; i++) {
        if (i == 0) {
            gt_num = a[i];
        } else {
            if (a[i] > gt_num) {
                gt_num = a[i];
            }
        }
        printf("Loop %d: gt_num = %d\n", i, gt_num);
    }

    return 0;
}
```

Solution:

```
Loop 0: gt_num = 42
Loop 1: gt_num = 42
Loop 2: gt_num = 44
Loop 3: gt_num = 51
Loop 4: gt_num = 51
Loop 5: gt_num = 51
Loop 6: gt_num = 92
Loop 7: gt_num = 92
Loop 8: gt_num = 92
Loop 9: gt_num = 100
```

12. (12 points) (BONUS) This question tests your understanding of C variables and operations. Your goal is to track the variable known as crewmate as it progresses through the code. Note the changes to its value, and write down what each of the `printf` statements will print out. Observe crewmate as it does its tasks!

```
1  #include <stdio.h>
2
3  int reactor (int a);
4  int admin (int b);
5  void navigation (int c);
6
7  int main() {
8      int crewmate = 304;
9      printf("Crewmate value is: %d\n", crewmate);
10
11     crewmate = reactor(crewmate);
12     printf("Crewmate value is: %d\n", crewmate);
13
14     crewmate = crewmate + (34 >> 1);
15     printf("Crewmate value is: %d\n", crewmate);
16
17     navigation(crewmate);
18     printf("Crewmate value is: %d\n", crewmate);
19
20     crewmate = crewmate + ((5 % 3) * 1);
21     printf("Crewmate value is: %d\n", crewmate);
22
23     crewmate = admin(crewmate);
24     printf("Crewmate value is: %d\n", crewmate);
25     printf("Congrats, you've completed all %d tasks!\n", crewmate);
26
27     return 0;
28 }
29
30 int reactor (int a) {
31     int meltdown = 3;
32     int manifolds = 1;
33
34     for (int i = 0; i < 3; i++) {
35         manifolds = (manifolds * meltdown) % 6;
36     }
37
38     int patterns = a + (manifolds - 2);
39     return patterns;
40 }
41
42 void navigation (int c) {
43     int steering = 1;
44
45     while ((steering ^ c) % 2 == 1) {
46         steering = (steering * 3) % 7;
```

```
47          if (steering == 1) {
48              break;
49          }
50      }
51  }
52
53  int admin (int b) {
54      int card = 10;
55      for (int swipe = 0; swipe < 10; swipe++) {
56          card += (card*2)%5;
57      }
58      if (card > 50) {
59          b = 150;
60          return b;
61      } else {
62          b = 140;
63          return b;
64      }
65  }
```

OUTPUT:

Solution:

```
Crewmate value is: 304
Crewmate value is: 305
Crewmate value is: 322
Crewmate value is: 322
Crewmate value is: 324
Crewmate value is: 140
Congratulations, you've completed all 140 tasks!
```