

ENEE 140, Spring 2015
Final Exam

Wednesday, May 20, 2015, 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: _____ Section: _____ Directory ID: _____

1 (16):
2 (10):
3 (12):
4 (8):
5 (15):
6 (5):
7 (16):
8 (18):
TOTAL (100):

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!

Problem 1. (16 points)

This problem tests your understanding of C types and C operations. Assume that variables a, b, c, d and e are defined as followed:

```
int      a = 1;
int      b = 3;
float    c = 1;
float    d = 2;
int      e = 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 r2 - r9 variables.

	Assignment	Value
float	r1 = c / d;	0.5
float	r2 = a / (int)d;	
int	r3 = a % b;	
unsigned	r4 = UINT_MAX % 2;	
int	r5 = a && b;	
int	r6 = e++;	
int	r7 = (a - (unsigned)b) > 0;	
float	r8 = (a / b > 0) ? c : d;	
int	r9 = b >> 1;	

Problem 2. (10 points)

This problem tests your understanding of assignments, equality tests and `if` statements. What is the output of the following code snippet?

```
int a = 5;
if (a = 0)
    printf("a=0 is true. \n");
else
    printf("a=0 is not true.\n");
if (a==0)
    printf("a==0 is true. \n");
else
    printf("a==0 is not true.\n");
```

Problem 3. (12 points)

This problem tests your understanding of control flow. Considering the following code segment, what will be the values of `i` and `k` when the program exits the second `for` loop?

```
for (i=0; i<=10; i=i+2) {
    if (i == 6)
        break;
    else i++;
}
```

```
for (k=0; k<=10; k=k+2) {
    if (k == 6)
        continue;
    else k++;
}
```

`i = _____ k = _____`

Problem 4. (8 points)

This program tests your understanding of strings. What is the output of the following code snippet? Write one character on each space, including the white space.

```
char s1[] = "ENEE 140 Final";
char s2[10] = "Exam";
s1[9] = '\\0';
printf("%s%s", s1, s2);
```


Problem 5. (15 points)

This problem tests your understanding of command line arguments and of the switch statement. Fill in the blanks in the program on the next page in order to implement the following functionality:

- If there is only one command line argument, print out "You did not provide any input."
- If there are two arguments, print out "You are signed up for [argument 2]."
- If there are three arguments, print out "You are signed up for [argument 2] in the [argument 3] semester."
- If there are more than three arguments, print out "Too many arguments!"

Example invocations:

Command line invocation: a.out

output: You did not provide any input.

Command line invocation: a.out ENEE140

Output: You are signed up for ENEE140

Command line input: a.out ENEE140 Spring

Output: You are signed up for ENEE140 in the Spring semester.

Command line input: a.out ENEE140 Spring 2015

Output: Too many arguments!

The program:

```
#include <stdio.h>

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

switch(_____)
{
case 1:
    printf("You did not provide any input.\n");
    _____;
case __:
    printf("You are signed up for %s.\n",_____);
    _____;
case __:
    printf("You are signed up for %s in the %s semester.\n",_____,_____);
    _____;
_____ :
    printf("Too many arguments!\n");
}

return 0;
}
```

Problem 6. (5 points)

This problem tests your understanding of multi-dimensional arrays. Assume you have the following 2D array called `data[8][8]`:

```
10101010
10101010
10101010
10101010
10101010
10101010
10101010
10101010
10101010
10101010
```

What would the value of variable `sum` be at the end of this segment of code?

```
for(i=0;i<8;i++){
    for(j=0;j<8;j++){
        if(i==j)
            sum=sum+data[i][j];
    }
}
```

Problem 7. (16 points)

This problem tests your understanding of characters and strings. Fill in the blanks of the following code to prompt the user for the string `first` and change the string `first` to all capital letters and print out the result. Assume all 8 spaces are filled with characters but not necessarily letters. If there is a non-letter in the string, ignore it and continue to the next character.

There should be 8 blanks to fill.

```
char first[____];
printf("What is your first name?\n");
scanf("%____",____);
int i;
for(i=0;i<____;i++) {
    if(first[i]>=____ && first[i] <=____) {
        first[i] = _____;
    }
}
printf("%____",first);
```

Problem 8. (15 points)

This problem tests your understanding of random numbers and control flow.

A. In the blank space from the code below, write a statement that generates a random odd number between 20–40 and assigns it to variable `r`:

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

int main()
{
    srand(time(NULL));

    int nums[10];
    int i,r,j,f;

    for (i=0;i<10;i++) {
        f=0;
        r=_____ ;           // <--- WRITE YOUR STATEMENT HERE
        if (i==0) {
            nums[i]=r;
        }
        else {
            for (j=0;j<i;j++) {
                if (nums[j]==r)
                    f=1;
            }
            if (f==0) {
                nums[i]=r;
            }
            else
                i--;
        }
    }

    for (i=0;i<10;i++)
        printf("%d -> %d\n",i,nums[i]);
}
```


B. State in plain English the purpose of this program. In your explanation, include the purpose of variable f .