

ME/ECE/EMA/CS 759
High Performance Computing for Engineering Applications
Assignment 1

1. The purpose of this exercise is to understand how to generate an executable on Euler, the cluster that we'll be using in ME759. To this end, write a very short program whose output is "Hello! I'm student WXYZ.", where WXYZ is the set of four least significant digits of your UW-Madison student ID. This ID is stored in an input file called `problem1.txt`. You should open this file, read the ID, and print the relevant digits. For instance, if I was student 0023903422, the output would be "Hello! I'm student 3422." You might want to use the `sprintf` function or friends to convert your ID to a string and then pick up the relevant four characters of the string. Keep in mind that some IDs are really long (an `int` won't hold them and you will have to use a different data type). We will test your code with other input files, each containing valid IDs. To test your solution, we will invoke the `makefile` and then type:

```
>> problem1.exe
```

If you develop under Windows or MacOS, make sure your code runs on Euler, which is a Linux machine.

2. Write a C program that reads a string provided as a command line argument. Pass the string to a function that you write. This function should count the total number of characters in the string excluding the null character. Return this value and print it out in the main program. We are going to test your program by passing it all sorts of strings: from empty strings, to strings that are 256 characters long. We won't pass it a string that is longer than that. To test your solution, we will invoke the `makefile` and then type something like:

```
>> problem2.exe "This is a test"
```

We'll expect to get the output (length of the string):

```
>> 14
```

3. The purpose of this problem is to learn how to use the `gdb` debugger under Euler and to understand better how pointer arithmetic works. To this end, you will have to use the flag `-g` when compiling your code with `g++` to include debug information in the executable. Consider the code in the text-box below. Use the `gdb` debugger to step through the code and answer the following questions:

- a) What is the size of variable `p` on Euler?
- b) What is the address of `p`?
- c) What is the address of `c`?
- d) What is the value of `arr[0]` after the assignment on line 16?
- e) What is the value of `arr[0]` at the end of the program?
- f) Explain why the value of `arr[0]` changes

To be able to use a grading script, your answer for the questions a through f should be provided each on one line in a file called `problem3.out`.

```

#include<iostream>

int main() {
    int d;
    char c;
    short s;
    int* p;
    int arr[2];

    p = &d;
    *p = 10;
    c = (char)1;

    p = arr;
    *(p+1) = 5;
    p[0] = d;

    *( (char*)p + 1 ) = c;

    return 0;
}

```

FURTHER NOTES:

- Provide solutions in your ME759 repo in the folder **HW01**. Therein, there should be a **makefile** which we will run. The output of this step is a set of executables: **problem1.exe**, **problem2.exe**, **problem3.exe**. We will run these programs to grade your work based on the output we get.
- On the due date, the time when the assignment is due is 11:59 PM. After 11:59 PM you will start burning time out of your five-day “late homework” budget.
- Use the class Canvas forum for exchanging ideas or sharing links to helpful information. No posting of the solutions please.
- This is an individual assignment. Please visit the first lecture to review what’s allowed and what’s not allowed in terms of using information from a third party