1. Write an assembly language program that prompts the user first for their first name and then for their last name and then displays “Hello” followed by their name. An example run of the program is shown below. (Program output is shown in bold.)

   Please enter your first name: Bill
   Please enter your last name: Rumpelstiltskin
   Hello Bill Rumpelstiltskin!

   Use the CS220 library routines for input and output.

2. Write an assembly language program that simulates 4 rolls of a die by displaying 4 random numbers between 1 and 6 inclusive. Call the C standard library function `rand()` (as illustrated in lecture) to generate the random number. Seed the random number generator using the current time so that successive runs of the program produce 4 new random numbers. You should only call `srand()` once (at the beginning of the program) to seed the generator. You will need to make 4 calls to `rand()`.

   The `rand()` function returns a positive random integer between 0 and RAND_MAX. (RAND_MAX is defined in the cstdlib header and is equal to 2,147,483,647 \(2^{31}-1\) or 7fffffff\(_{16}\) under MinGW and Linux.) In C++ we could use `rand()` to generate a random number between 1 and 6 as follows:

   ```
   die = rand() % 6 + 1;
   ```

   where \% is the remainder (modulo) operator. `rand() % 6` yields a number between 0 and 5. Adding 1 then gives a number between 1 and 6. In assembly the instruction:

   ```
   idivl reg
   ```

   where reg is one of the general purpose registers will divide the 64 bit number stored in EDX:EAX by the number in reg. The quotient is stored in EAX and the remainder in EDX. Use the instruction

   ```
   incl reg
   ```

   to increment a register by one.

   Write your program so that no values need to be stored in either the data or bss sections, that is, keep all values stored in the microprocessor registers.

   You are not expected to do this is a loop. Figure out how to generate and display one random number and then duplicate that section of code 3 times to display 4 random numbers.