CSCI 210 Assn. 3: Multi-argument subroutines

This assignment is due at 4pm, Friday, February 20. Note that there is no paper submission for this assignment; you should submit your solution electronically.

You may work with another student if you like and submit an answer with that student. (Include a comment at the beginning of your submitted file mentioning the names of both students.)

Write a subroutine `max` in the x86 assembly language. The number of arguments sent to the subroutine is variable: The first argument (an integer) says how many more integer arguments there are, and the function should return the maximum among the following arguments. For example, if a program were to call “`max(5, 3, 2, 4, 1, 0)`”, the subroutine would return 4: The 5 indicates how many more arguments there are, and the maximum among these five (3, 2, 4, 1, and 0) is 4.

Running “`getcs 210 4b`” will fetch three files.

`test.c` is for testing your program. It illustrates how your program might be used.

```
#include <stdio.h>

int max(int n, ...);

int main() {
    int i, j, k, m, n;

    printf("Enter five numbers: ");
    scanf("%d%d%d%d%d", &i, &j, &k, &m, &n);
    printf("Max of first three: %d\n", max(3, i, j, k));
    printf("Max of all five: %d\n", max(5, i, j, k, m, n));
    return 0;
}
```

`debug.s` contains two subroutines that are useful for debugging assembly code: printregs prints the current register values, and printstack prints the top eight 32-bit values on the stack (not including the return address pushed by call printstack). Both subroutines take no arguments, and they do not make permanent changes to any registers, including the caller-save registers.

`max.s` is a template into which you will place your code.

```
.globl max
.section .text
max:    pushl %ebp          # entry template
        movl %esp, %ebp

        # your code here

        movl %ebp, %esp     # exit template
        popl %ebp
        ret
```

You can compile and run your program as follows.

```
chomas% gcc *.s *.c
chomas% ./a.out
```

You can submit your solution with the command “`handincs 210 4b`”. Your assembly code should be commented appropriately.