

## CS 261 (Computer Systems I) – Y86-64 Lab

Submit on stu using the normal project submission script (it will be project “y1”). Put the names of all your group members in a one-line comment at the top of your files beginning with “# Name:”. Only one person per group should submit the files. This lab is due on Friday, Oct 21.

### Part 1 – Basic Arithmetic

Write a Y86-64 program called “arith.y” that will calculate the answers to the following arithmetic problems and put the results in the specified registers:

3 + 4  
(%rcx)

24 - 8  
(%rbx)

3 + 4 - 5  
(%rbp)

### Part 2 – Loops

Write a Y86-64 program called “fib.y” that calculates the first 10 Fibonacci numbers (1, 1, 2, 3, etc.) using a loop. The numbers should be written to 8-byte slots in memory starting at location 0x400.

### Part 3 – Functions

Write a Y86-64 program called “sum.y” that is equivalent to the following C code. You must implement x86 calling conventions for sum() with parameters in %rdi and %rsi (respectively) and a return value in %rax. Note that this means that sum() must save/restore %rbx and %rbp if they are used.

```
int64_t nums[] = { 1, 2, 3, 4 };
int64_t len = 4;

int64_t main (void)
{
    return sum (nums, len);
}

int64_t sum (int64_t *array, int64_t size)
{
    int64_t result = 0;
    while (size > 0) {
        result += *array;
        array++;
        size--;
    }
    return result;
}
```

The global array and its length should be stored in a data segment as follows:

```
.pos 0x300 data
len:
    .quad 4
nums:
    .quad 1
    .quad 2
    .quad 3
    .quad 4
```