# Lab 4

Introduction to Programming Laboratory

# Goals

- HW1 Commentary
- HW2 Overview
- Tutorial: Finding memory problems
- OpenMP
- Task: Primes - Composites

# HW1 Commentary

# HW2 Overview

# Tutorial: Finding memory problems

## Summary

Just add `-g -Og -fsanitize=address` to the compiler and run your code normally.

See the cheatsheet for more information.

OpenMP

# #pragma omp parallel for

Parallelize this for loop

```
#pragma omp parallel for
for (int i = 0; i < jobs; i++)
```

## scheduling options

- schedule(static) `aaaaa|bbbbb|ccccc`
- schedule(static, 1) `a|b|c|a|b|c|a|b|c|a|b|c|a|b|c`
- schedule(dynamic, 1) `a|b|c|a|c|c|a|a|b|b|a|b|a|b|c`
- schedule(dynamic, 2) `aa|bb|cc|cc|aa|aa|bb|b`
- schedule(guided)

## Compile

Add `-fopenmp`

## Run

Add `-c#` to `srun`, where `#` is the number of CPUs per process.

# Task: Primes - Composites

Given a number N, find out the primes and composite numbers ≤ N

Calculate sum(primes) - sum(composites)

## Requirements

- Start with the given sequential code at `/home/ipl19/x/lab4/lab4seq.c`
- Use MPI and OpenMP to parallelize.
- Use `-fsanitize=address` to check for memory errors.
- Name your source code `lab4.c` or `lab4.cc`
- Your code must be within 50 lines.
- Run `x` and come to TA.