CSCI 455: Lab #2 — Parallel Sum

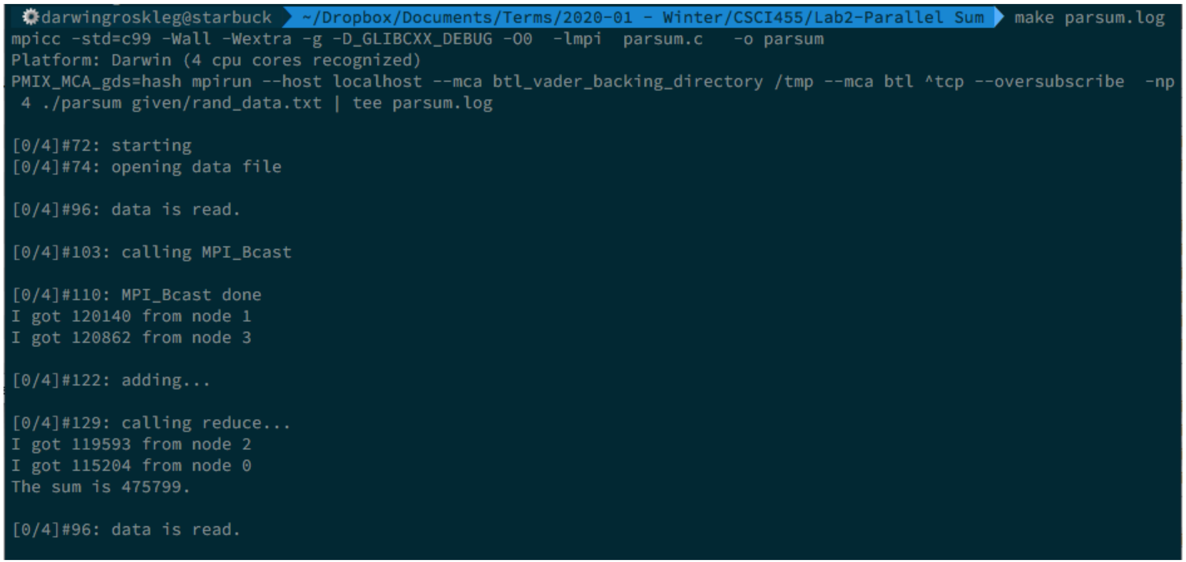
Darwin Jacob Groskleg

Winter 2020

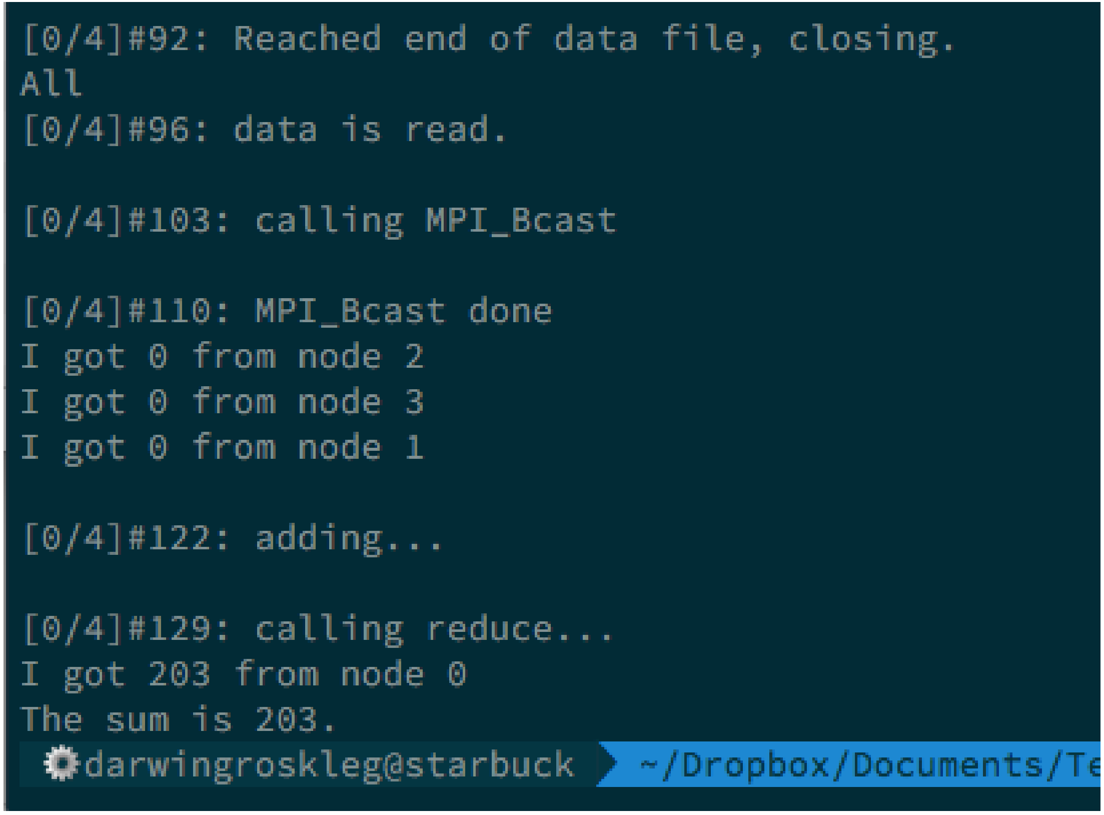
Contents

Parallel Sum

Program Outputs and Results



```
darwingroskleg@starbuck  ~/Dropbox/Documents/Terms/2020-01 - Winter/CSCI455/Lab2-Parallel Sum  make parsum.log
mpicc -std=c99 -Wall -Wextra -g -D_GLIBCXX_DEBUG -O0  -lmpi  parsum.c   -o parsum
Platform: Darwin (4 cpu cores recognized)
PMIX_MCA_gds=hash mpirun --host localhost --mca btl_vader_backing_directory /tmp --mca btl ^tcp --oversubscribe  -np
 4 ./parsum given/rand_data.txt | tee parsum.log

[0/4]#72: starting
[0/4]#74: opening data file

[0/4]#96: data is read.

[0/4]#103: calling MPI_Bcast

[0/4]#110: MPI_Bcast done
I got 120140 from node 1
I got 120862 from node 3

[0/4]#122: adding...

[0/4]#129: calling reduce...
I got 119593 from node 2
I got 115204 from node 0
The sum is 475799.

[0/4]#96: data is read.
```



```
[0/4]#92: Reached end of data file, closing.
All
[0/4]#96: data is read.

[0/4]#103: calling MPI_Bcast

[0/4]#110: MPI_Bcast done
I got 0 from node 2
I got 0 from node 3
I got 0 from node 1

[0/4]#122: adding...

[0/4]#129: calling reduce...
I got 203 from node 0
The sum is 203.
 darwingroskleg@starbuck  ~/Dropbox/Documents/Te
```

```
I got 120140 from node 1
I got 120862 from node 3
I got 119593 from node 2
I got 115204 from node 0
```

```
The sum is 475799.
I got 123625 from node 2
I got 128131 from node 3
I got 126174 from node 0
I got 121279 from node 1
The sum is 499209.
I got 125587 from node 0
I got 127182 from node 1
I got 130344 from node 2
I got 131021 from node 3
The sum is 514134.
I got 136878 from node 0
I got 128938 from node 1
I got 128562 from node 3
The sum is 519745.
I got 125367 from node 2
I got 124628 from node 2
I got 126732 from node 3
I got 126149 from node 0
I got 126446 from node 1
The sum is 503955.
I got 120813 from node 1
I got 133427 from node 2
I got 117295 from node 3
I got 126377 from node 0
The sum is 497912.
I got 125013 from node 0
I got 127937 from node 1
I got 118481 from node 2
I got 127119 from node 3
The sum is 498550.
I got 122837 from node 0
I got 122432 from node 1
I got 132499 from node 2
I got 126747 from node 3
The sum is 504515.
I got 119981 from node 0
I got 127754 from node 1
I got 131788 from node 2
I got 122037 from node 3
The sum is 501560.
I got 131305 from node 0
I got 118236 from node 3
I got 121594 from node 1
I got 120244 from node 2
The sum is 491379.
```

```
I got 113915 from node 0
I got 129497 from node 2
I got 130283 from node 3
I got 132475 from node 1
The sum is 506170.
I got 124031 from node 0
I got 130685 from node 1
I got 127476 from node 2
I got 125216 from node 3
The sum is 507408.
I got 123962 from node 3
I got 123973 from node 0
I got 130347 from node 1
I got 119931 from node 2
The sum is 498213.
I got 123585 from node 0
I got 117865 from node 1
I got 122527 from node 2
I got 116444 from node 3
The sum is 480421.
I got 123761 from node 0
I got 119529 from node 1
I got 124949 from node 2
The sum is 481882.
I got 113643 from node 3
I got 116309 from node 3
I got 130241 from node 0
I got 128377 from node 1
I got 123402 from node 2
The sum is 498329.
I got 125721 from node 0
I got 126364 from node 1
I got 134281 from node 2
I got 127600 from node 3
The sum is 513966.
I got 124134 from node 0
I got 125723 from node 1
I got 118079 from node 2
I got 120106 from node 3
The sum is 488042.
I got 133677 from node 2
I got 128268 from node 0
I got 133150 from node 3
I got 123621 from node 1
The sum is 518716.
I got 127992 from node 1
```

```
I got 131721 from node 2
I got 127557 from node 0
I got 126205 from node 3
The sum is 513475.
I got 120132 from node 0
I got 119738 from node 1
I got 132315 from node 2
I got 132782 from node 3
The sum is 504967.
I got 127185 from node 0
I got 126061 from node 1
I got 126975 from node 2
I got 122329 from node 3
The sum is 502550.
I got 124104 from node 1
I got 127542 from node 3
I got 125520 from node 0
I got 131855 from node 2
The sum is 509021.
I got 129359 from node 0
I got 119845 from node 1
I got 125165 from node 2
I got 120846 from node 3
The sum is 495215.
I got 111428 from node 0
I got 126243 from node 1
I got 125859 from node 2
I got 121184 from node 3
The sum is 484714.
I got 125290 from node 0
I got 121607 from node 2
I got 126579 from node 1
I got 129976 from node 3
The sum is 503452.
I got 121952 from node 2
I got 129872 from node 0
I got 128131 from node 1
I got 130429 from node 3
The sum is 510384.
I got 127343 from node 0
I got 122040 from node 1
I got 128792 from node 2
I got 125022 from node 3
The sum is 503197.
I got 119139 from node 0
I got 124425 from node 1
```

```
I got 128279 from node 3
I got 128066 from node 2
The sum is 499909.
I got 130295 from node 0
I got 121802 from node 1
I got 125692 from node 2
I got 128718 from node 3
The sum is 506507.
I got 121695 from node 1
I got 127757 from node 2
I got 123029 from node 0
I got 124188 from node 3
The sum is 496669.
I got 121544 from node 0
I got 133499 from node 1
I got 117988 from node 2
I got 124333 from node 3
The sum is 497364.
I got 123114 from node 0
I got 125384 from node 2
I got 118681 from node 1
I got 122362 from node 3
The sum is 489541.
I got 116260 from node 2
I got 126050 from node 3
I got 122395 from node 0
I got 121070 from node 1
The sum is 485775.
I got 121564 from node 1
I got 132977 from node 0
I got 126379 from node 2
I got 118014 from node 3
The sum is 498934.
I got 127285 from node 0
I got 128974 from node 1
I got 124388 from node 2
I got 122159 from node 3
The sum is 502806.
I got 127502 from node 0
I got 120194 from node 1
I got 122545 from node 2
I got 130545 from node 3
The sum is 500786.
I got 130051 from node 0
I got 127513 from node 1
I got 129636 from node 2
```

```
I got 125291 from node 3
The sum is 512491.
I got 128048 from node 0
I got 123337 from node 1
I got 123227 from node 2
I got 131905 from node 3
The sum is 506517.
I got 124015 from node 1
I got 120024 from node 2
I got 118723 from node 3
I got 128188 from node 0
The sum is 490950.
I got 122970 from node 1
I got 130631 from node 2
I got 125749 from node 3
I got 115772 from node 0
The sum is 495122.
I got 126071 from node 0
I got 128101 from node 1
I got 124489 from node 2
I got 121572 from node 3
The sum is 500233.
I got 121855 from node 0
I got 133435 from node 1
I got 119041 from node 2
I got 126711 from node 3
The sum is 501042.
I got 130226 from node 0
I got 125967 from node 1
I got 130191 from node 2
I got 136065 from node 3
The sum is 522449.
I got 122682 from node 3
I got 126714 from node 0
I got 128810 from node 1
I got 122044 from node 2
The sum is 500250.
I got 117577 from node 0
I got 121599 from node 1
I got 131805 from node 3
I got 125760 from node 2
The sum is 496741.
I got 123080 from node 1
I got 130833 from node 2
I got 128062 from node 3
I got 118327 from node 0
```

```
The sum is 500302.
I got 122017 from node 3
I got 123669 from node 0
I got 124022 from node 1
I got 128503 from node 2
The sum is 498211.
I got 121428 from node 0
I got 118372 from node 1
I got 128445 from node 2
I got 121926 from node 3
The sum is 490171.
I got 132589 from node 1
I got 130964 from node 0
I got 120718 from node 2
I got 119760 from node 3
The sum is 504031.
I got 122739 from node 0
I got 123293 from node 1
I got 122971 from node 2
I got 119394 from node 3
The sum is 488397.
I got 125170 from node 1
I got 127886 from node 2
I got 128225 from node 3
I got 123071 from node 0
The sum is 504352.
I got 121810 from node 3
I got 129308 from node 0
I got 128089 from node 1
I got 126861 from node 2
The sum is 506068.
I got 126996 from node 0
I got 132467 from node 1
I got 130965 from node 3
I got 114974 from node 2
The sum is 505402.
I got 126626 from node 0
I got 125941 from node 1
I got 123138 from node 2
I got 121989 from node 3
The sum is 497694.
I got 121351 from node 0
I got 124207 from node 1
I got 129603 from node 2
I got 124155 from node 3
The sum is 499316.
```

```
I got 130198 from node 1
I got 129623 from node 2
I got 127500 from node 3
I got 125052 from node 0
The sum is 512373.
I got 119446 from node 1
I got 121399 from node 2
I got 127822 from node 3
I got 120690 from node 0
The sum is 489357.
I got 133890 from node 0
I got 129673 from node 1
I got 132280 from node 2
I got 119906 from node 3
The sum is 515749.
I got 122703 from node 0
I got 129744 from node 1
I got 132177 from node 2
I got 123091 from node 3
The sum is 507715.
I got 127348 from node 2
I got 122991 from node 3
I got 132784 from node 1
I got 124735 from node 0
The sum is 507858.
I got 129437 from node 0
I got 122535 from node 1
I got 125219 from node 2
I got 126486 from node 3
The sum is 503677.
I got 128700 from node 2
I got 122042 from node 3
I got 126700 from node 0
I got 118957 from node 1
The sum is 496399.
I got 125506 from node 2
I got 126637 from node 0
I got 124232 from node 1
I got 119127 from node 3
The sum is 495502.
I got 123674 from node 0
I got 126733 from node 1
I got 120991 from node 2
I got 117073 from node 3
The sum is 488471.
I got 125890 from node 0
```

```
I got 124565 from node 1
I got 124976 from node 2
I got 132982 from node 3
The sum is 508413.
I got 124560 from node 0
I got 132099 from node 1
I got 119999 from node 2
I got 125089 from node 3
The sum is 501747.
I got 123990 from node 0
I got 116861 from node 1
I got 121765 from node 2
I got 120165 from node 3
The sum is 482781.
I got 122349 from node 3
I got 122538 from node 0
I got 123720 from node 1
I got 118804 from node 2
The sum is 487411.
I got 117475 from node 2
I got 116051 from node 0
I got 123196 from node 1
I got 128110 from node 3
The sum is 484832.
I got 131305 from node 0
I got 134177 from node 1
I got 123091 from node 2
I got 123572 from node 3
The sum is 512145.
I got 123890 from node 0
I got 127426 from node 2
I got 120906 from node 3
I got 119731 from node 1
The sum is 491953.
I got 118288 from node 0
I got 123247 from node 1
I got 125360 from node 2
The sum is 489120.
I got 122225 from node 3
I got 119878 from node 2
I got 120190 from node 0
I got 124876 from node 1
I got 132493 from node 3
The sum is 497437.
I got 122019 from node 0
I got 138115 from node 1
```

```
I got 125794 from node 3
I got 137214 from node 2
The sum is 523142.
I got 119792 from node 0
I got 122666 from node 1
I got 129311 from node 2
I got 121901 from node 3
The sum is 493670.
I got 132640 from node 1
I got 130966 from node 3
I got 120207 from node 0
I got 120544 from node 2
The sum is 504357.
I got 121653 from node 2
I got 120937 from node 0
I got 121882 from node 1
I got 123551 from node 3
The sum is 488023.
I got 123168 from node 0
I got 127624 from node 1
I got 122890 from node 2
I got 119915 from node 3
The sum is 493597.
I got 129529 from node 2
I got 122481 from node 0
I got 137370 from node 1
The sum is 506854.
I got 117474 from node 3
I got 125904 from node 0
I got 127036 from node 1
I got 133553 from node 2
I got 114582 from node 3
The sum is 501075.
I got 115201 from node 0
I got 127288 from node 1
I got 120560 from node 2
I got 126537 from node 3
The sum is 489586.
I got 122491 from node 0
I got 123247 from node 1
I got 119979 from node 2
I got 123860 from node 3
The sum is 489577.
I got 119329 from node 1
I got 115520 from node 2
I got 127775 from node 3
```

```
I got 119022 from node 0
The sum is 481646.
I got 129941 from node 3
I got 118269 from node 0
I got 119354 from node 1
I got 123895 from node 2
The sum is 491459.
I got 125982 from node 1
I got 131692 from node 2
I got 130529 from node 3
I got 127239 from node 0
The sum is 515442.
I got 114332 from node 1
I got 127072 from node 0
I got 121655 from node 2
I got 130427 from node 3
The sum is 493486.
I got 120784 from node 3
I got 123820 from node 0
I got 122834 from node 1
I got 127033 from node 2
The sum is 494471.
I got 122715 from node 3
I got 130366 from node 1
I got 126390 from node 2
I got 132521 from node 0
The sum is 511992.
I got 115460 from node 2
I got 129377 from node 3
I got 128591 from node 0
I got 120700 from node 1
The sum is 494128.
I got 128038 from node 1
I got 122269 from node 0
The sum is 506846.
I got 128369 from node 2
I got 128170 from node 3
I got 125545 from node 0
I got 119999 from node 1
I got 112101 from node 2
I got 127263 from node 3
The sum is 484908.
I got 132638 from node 1
I got 123165 from node 2
I got 127375 from node 3
I got 130202 from node 0
```

```
The sum is 513380.
I got 122936 from node 1
I got 125092 from node 0
I got 119556 from node 2
I got 118760 from node 3
The sum is 486344.
I got 121768 from node 0
I got 128948 from node 2
I got 127466 from node 3
The sum is 508533.
I got 130351 from node 1
I got 125249 from node 0
I got 125765 from node 1
I got 119106 from node 2
I got 124806 from node 3
The sum is 494926.
I got 128554 from node 1
I got 131674 from node 2
I got 123149 from node 0
The sum is 505607.
I got 122230 from node 3
I got 124947 from node 0
I got 127026 from node 1
I got 126011 from node 2
I got 123196 from node 3
The sum is 501180.
I got 130524 from node 3
I got 127033 from node 1
I got 124449 from node 2
I got 118569 from node 0
The sum is 500575.
I got 123847 from node 1
I got 134733 from node 2
I got 122251 from node 0
I got 119827 from node 3
The sum is 500658.
I got 0 from node 2
I got 0 from node 3
I got 0 from node 1
I got 203 from node 0
The sum is 203.
```

parsum.c

```c
/* parsum.c
 * ---------
 * Authors: Darwin Jacob Groskleg, Laurence T. Yang
 * CSCI 455
 * Lab #2: Parallel Sum
 *
 * Purpose: compute the sum of numbers read in from a file by distributing the
 *          computation in blocks of 100 numbers over the processors.
 *
 *          Uses the ”Broadcast/Reduce Routine”, see Lecture 10, for
 *          partitioning the summation.
 *
 * TODO:
 *  - [x] Fill in all missing parts of the code.
 *  - [x] Should find out the number of processors in the cluster and divide the
 *        work evenly among them instead of assuming it has 10 processors doing
 *        100 numbers each.
 *  - [ ] Modify the program so that it also computes the MIN and MAX values of
 *        the data read in.
 *        Tips: you'll write your own min/max routines that find the min/max of
 *        each block of numbers, and then have a call to MPI reduce that finds
 *        the min/max of all the ”partial” results.
 */
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>
#include <stdbool.h>

#include <mpi.h>
#include ”trace.h”

#define MAXSIZE 1000  // original limit
//#define MAXSIZE 100000
#define MAX(a,b) ((a) > (b) ? (a) : (b))
#define MIN(a,b) ((a) < (b) ? (a) : (b))

/* Initialize with a limit state. */
typedef struct {
    int sum;
    int min;
    int max;
} range_sum_t;

range_sum_t minmaxadd(int *A, int low, int high);
int add(int *A, int low, int high);
int sign(int z);
int div_out(int num, int denom);
void show_usage(char* program);

int myid, cluster_size;

int main(int argc, char *argv[]) {
    /* 4 Synchronized variables, requiring broadcasts from root */
    int set_size = 0;
    int data[MAXSIZE];
    int max_partition_size;
    bool file_contains_data = true;

    MPI_Init(&argc, &argv);                         /* Initialize MPI */
    MPI_Comm_size(MPI_COMM_WORLD, &cluster_size);   /* Find group size */
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);           /* Find process rank */
    bool root_process = (myid == 0);

    /* Root: open the data file */
    char* datafilename = argv[1];
    FILE *fp;   /* file pointer, initialized by fopen */
    if (root_process) {
        if (argc != 2)
            show_usage(argv[0]);
        TRACE(”starting”);

        TRACE(”opening data file \n”);
        /* Open Input File and Initialize Data Array */
        if ((fp = fopen(datafilename,”r”)) == NULL) {
            printf(”Can't open the input file: %s\n\n”, datafilename);
            MPI_Abort(MPI_COMM_WORLD, 1);
            exit(1);
        }
    }

    /* only important to root */
    long long global_sum = 0;
    while (file_contains_data) {
        /* Root Proc: copy numbers to an array from open data file. */
        if (root_process) {
            for(set_size=0; set_size<MAXSIZE && !feof(fp); set_size++)
                fscanf(fp,”%d”, &data[set_size]);

            if (feof(fp)) {
                TRACE(”Reached end of data file, closing.\nAll ”);
                fclose(fp);
                file_contains_data = false;
            }
            TRACE(”data is read.\n”);
            max_partition_size = div_out(set_size, cluster_size);
        }


        /* Broadcast the partition size and data to all node processes */
        if (root_process)
            TRACE(”calling MPI_Bcast\n”);
        MPI_Bcast( &file_contains_data,        1, MPI_INT, 0, MPI_COMM_WORLD);
        MPI_Bcast( &set_size,                  1, MPI_INT, 0, MPI_COMM_WORLD);
        MPI_Bcast( &max_partition_size,        1, MPI_INT, 0, MPI_COMM_WORLD);
        /* Bcast data[] */
        MPI_Bcast(               &data, set_size, MPI_INT, 0, MPI_COMM_WORLD);
        if (root_process)
            TRACE(”MPI_Bcast done \n”);


        /* Do the calculating: sum and reduce */
        MPI_Barrier(MPI_COMM_WORLD);
        /* Calculate the low and high index for each processor */
        int low_i, high_i;
        low_i  = myid * max_partition_size;
        high_i = MIN(low_i + max_partition_size, set_size) - 1;

        /* Local addition for all processes */
        if (root_process)
            TRACE(”adding...\n”);
        long long node_sum = add(data, low_i, high_i);
        printf(”I got %lld from node %d\n”, node_sum, myid);
        MPI_Barrier(MPI_COMM_WORLD);

        /* Global reduce */
        if (root_process)
            TRACE(”calling reduce...\n”);
        /* MPI_Reduce(
         *     void* send_data,
         *     void* recv_data,
         *     int count,
         *     MPI_Datatype datatype,
         *     MPI_Op op,
         *     int root,
         *     MPI_Comm communicator)
         */
        MPI_Reduce( // Sum operation
                &node_sum,
                &global_sum,    // receiver
                1,  // count of elements in the recv/send arrays
                MPI_LONG_LONG,
                MPI_SUM,
                0,  // root rank
                MPI_COMM_WORLD);
        if (root_process)
            printf(”The sum is %lld.\n”, global_sum);
    }

    /* MPI finalize */
    MPI_Finalize();
    return EXIT_SUCCESS;
}


void show_usage(char* program) {
    printf(”Usage: %s path/to/datafile.txt\n”, program);
    MPI_Abort(MPI_COMM_WORLD, 1);
    exit(1);
}

range_sum_t minmaxadd(int *A, int low, int high) {
    range_sum_t res = {0, INT_MAX, INT_MIN};
    for (int i=low; i<=high; i++) {
        res.sum += A[i];
        res.min = MIN(res.min, A[i]);
```

```c
        res.max = MAX(res.max, A[i]);
    }
    return res;
}

/* Adds range within an array,
 *  between positions low and high, inclusive.
 */
int add(int *A, int low, int high) {
    int res = 0;
    for (int i=low; i<=high; i++)
        res += A[i];
    return res;
}

/*
 * Give me a sign!
 *  Range: (-1, 0, 1)
 */
int sign(int z) {
    return ((0 < z) - (z < 0));
}

/* Integer division '/', in C99 and up, truncates towards zero.
 * We want to round away from zero so that we can ___
 * when deciding how many numbers to give to each node for summation.
 *
 *  int div in VS. int div out
 */
int div_out(int num, int denom) {
    div_t divresult = div(num, denom);
    return num/denom + sign(divresult.rem);
}
```

seqsum.c

```c
/* seqsum.c
 * ---------
 * Authors: Darwin Jacob Groskleg
 *
 * Sums up numbers in a given file sequentially.
 */
#include <stdio.h>
#include <stdlib.h>

void show_usage(char* pgm) {
    printf(”Usage: %s path/to/datafile.txt\n”, pgm);
    exit(1);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        show_usage(argv[0]);
    }

    //char* datafilename = ”given/rand_data.txt”;
    char* datafilename = argv[1];
    FILE *fp;
    if ((fp = fopen(datafilename,”r”)) == NULL) {
        printf(”Can't open the input file: %s\n\n”, datafilename);
        exit(1);
    }

    int z, count = 0;
    long long running_sum = 0;
    fscanf(fp, ”%d”, &z);
    while (!feof(fp)) {
        count++;
        running_sum += z;
        fscanf(fp, ”%d”, &z);
    }
    fclose(fp);

    printf(”%s contains %d numbers.\n”, datafilename, count);
    printf(”Global sum: %lld\n”, running_sum);

    return 0;
}
```

trace.h

```
/* trace.h
 * -------
 * Authors: Darwin Jacob Groskleg, Laurence T. Yang
 * CSCI 455
 * Lab #2: Parallel Sum
 */
#ifndef TRACE_H_INCLUDED
#define TRACE_H_INCLUDED

#include <stdio.h>

/* on Windows we define debug mode to be when _DEBUG is set */
#ifdef _DEBUG
#define DEBUG_MODE 1
#endif

/* on UNIX we define debug mode to be when _GLIBCXX is set */
#ifdef _GLIBCXX_DEBUG
#define DEBUG_MODE 1
#endif

#ifdef DEBUG
#define DEBUG_MODE 1
#endif


/* Our assertion macros do nothing in production. */
#ifndef DEBUG_MODE

#define TRACE( X )
#define TRACEd(X, V)

#else /* DEBUG_MODE */

#define TRACE( X ) \
    fprintf(stderr,”\n[%d/%d]#%d: %s”,myid,cluster_size,__LINE__,X)
#define TRACEd(X, V) \
    fprintf(stderr,”\n[%d/%d]#%d: %s%d”,myid,cluster_size,__LINE__,X,V)

#endif /* DEBUG_MODE */


#endif /* TRACE_H_INCLUDED */
```