CS 359
Parallel Computing
Assignment - 2

Name: Somya Mehta
Roll No.: 190001058

Problem Statement

Parallel merge sort starts with n/comm_size keys assigned to each process. It ends with all the keys stored on process 0 in sorted order. To achieve this, it uses the same tree-structured communication that we used to implement a global sum. However, when a process receives another process' keys, it merges the new keys into its already sorted list of keys. Write a program that implements parallel mergesort. Process 0 should read in n and broadcast it to the other processes. Each process should use a random number generator to create a local list of n/comm_size ints. Each process should then sort its local list, and process 0 should gather and print the local lists. Then the processes should use tree-structured communication to merge the global list onto process 0, which prints the result.

Solution Approach

I have solved the problem by dividing it into 2 sub-tasks. The first subtask we will divide the subarray into parts on which each process will work upon from process 0 to all the processes. Then, each process will randomly

generate an array of given size and sort it. After sorting, all the processes will return their respective arrays and here our 2nd subtask is there which we will solve i.e to form a tree-like hierarchy(binary tree) to merge the sub-arrays associated with each process.Now we print the array. This is similar to the parallel program used to sum all the numbers in an array.

Algorithm

Now, for the first subtask, these are the steps

1. Process 0 read in n and broadcast it to all processes since each process will handle this much size of sub-array.
2. Each process should use a random number generator to create a local list of n/comm size ints.
3. Each process should then sort its list.
4. Next, each process will return it's sorted subarray to process 0. Since the last processor may have a sub-array of different size, we have to use MPI_Gatherv to receive all the subarrays from all processes in a temporary array in process 0.
6. Lastly, Process 0 will output all the received subarrays. So, we

can represent the first subtask in the form of a (n-1)-ary tree.

Now, for the second subtask, we will have to merge the sorted subarrays.

So, if we are at process number i then we will first check if the current process has any child. The possible children of process i will be (2*i+1) and (2*i+2). To check if these children exist, we will simply compare with np. If the value is less than np then the corresponding process exists. Similarly, each process i except process 0 will have a parent and it's parent's process ID will be (i-1)/2. So what we will do is :->

1. Parent will receive the sorted subarrays from its children and merge them into an intermediate array and then this intermediate subarray will be merged with parent's own sub-array into a final array.
2. This is how each process will send the final array to its own parent.

Since process 0 is the parent of all so to ensure that process 0 outputs the final array only after all the merging steps are done, we have to use MPI_Barrier at the end before printing.

CODE

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>


int npes;
// here we merge the array with 2 pointers
int* merge(int* finalarr, int* a1, int *a2, int size1, int size2){
   int i=0, j=0, l=0;
   while(i < size1 && j < size2){
       if(a1[i] < a2[j]){
           finalarr[l] = a1[i];
           l++;
           i++;
       }
       else{
           finalarr[l] = a2[j];
           l++;
           j++;
       }
   }
   while(i < size1){
       finalarr[l] = a1[i];
       l++;
       i++;
   }
```

```
   while(j < size2){
       finalarr[l] = a2[j];
       l++;
       j++;
   }
   return finalarr;
}

int comparator (const void * x, const void * y) {
  return ( *(int*)x - *(int*)y );
}

int main(int argc, char **argv) {
   int myRank, n = 0, nelements = 0;

   MPI_Status status;
   MPI_Init (&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &myRank);
   MPI_Comm_size(MPI_COMM_WORLD, &npes);

   if(myRank == 0){
       // Input of number of elements
       printf("Enter Size of Array: ");
       scanf("%d", &n);
       nelements = n / npes;
   }
   int temp[n];

   // Broadcast to all processes
   MPI_Bcast(&nelements, 1, MPI_INT, 0, MPI_COMM_WORLD);
```

```
    // Remaining elements sent to last process
    if(myRank == 0){
        int last = n % npes;
        MPI_Send(&last, 1, MPI_INT, npes-1, 0, MPI_COMM_WORLD);
    }
    if(myRank == npes - 1){
        int extra;
        MPI_Recv(&extra, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, &status);
        nelements += extra;
    }

    // Generating Random Data
    int arr[nelements];

    int pid= myRank;
    srand(pid + 1);      // To make different arrays for different processes
    for(int i=0; i<nelements; i++){
        arr[i] = rand()%10000 + 1;
    }

    // the data is being sorted
    qsort(arr, nelements, sizeof(int), comparator);

    // Storing the lengths of arrays to be used for Gatherv
    int *recvlen = NULL;

    // if the base process is there then this much npes length we will
recieve
    if(myRank == 0) recvlen = malloc(npes * sizeof(int));
```

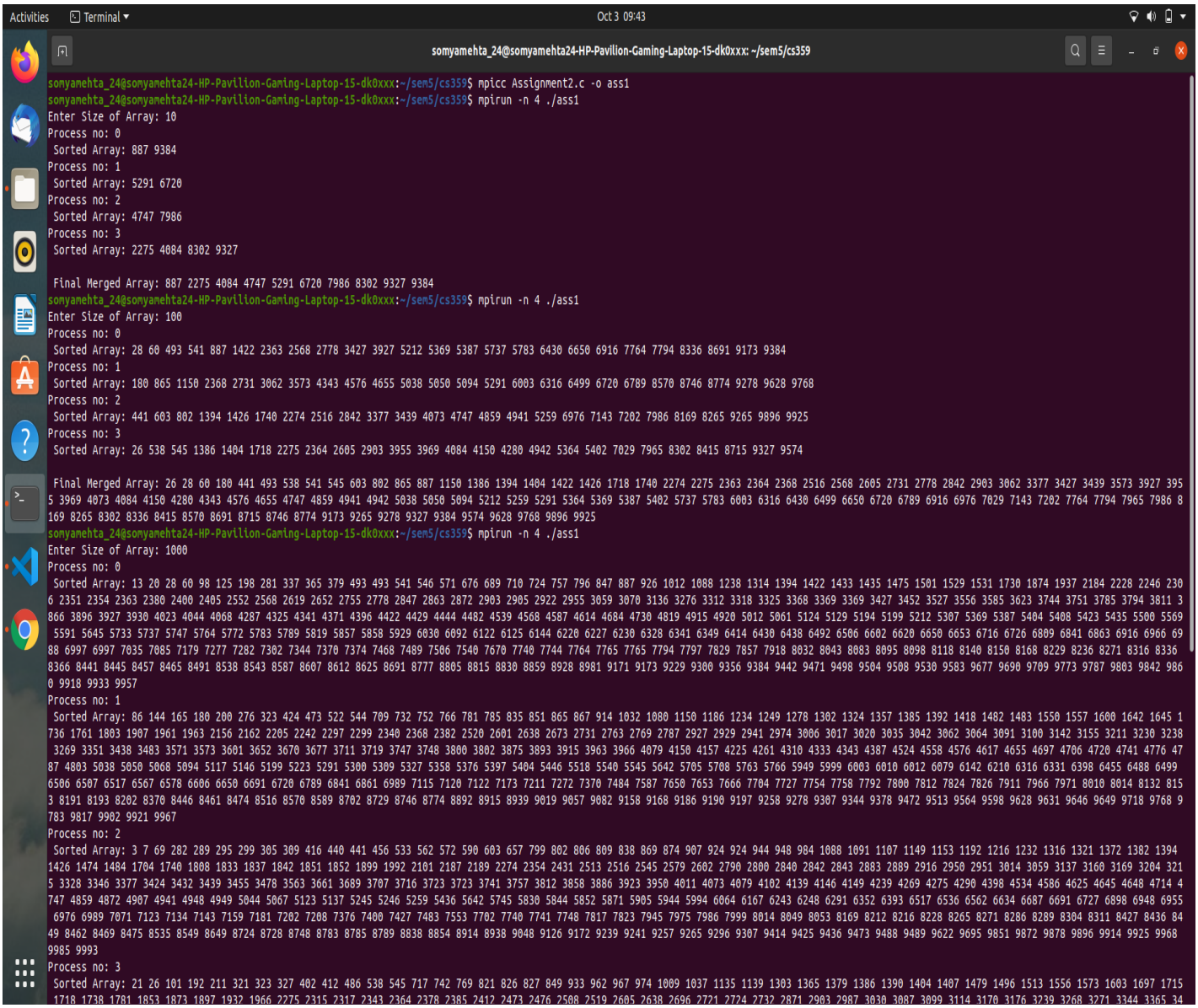
```
   MPI_Gather(&nelements, 1, MPI_INT, recvlen, 1, MPI_INT, 0,
MPI_COMM_WORLD);

   // Make array of displacements so that the recieve length of the
processes
   // can be stored  to be used in Gatherv later
   int *disp = NULL;

   if(myRank == 0){
       disp = malloc(npes * sizeof(int));
       disp[0] = 0;
       for(int i=1; i < npes; i++) disp[i] = disp[i-1] + recvlen[i-1];
   }

   // Gather all sorted sub-lists in temp array of process 0
   MPI_Gatherv(arr, nelements, MPI_INT, temp, recvlen, disp, MPI_INT, 0,
MPI_COMM_WORLD);

   // Print the sorted lists
   if(myRank == 0){
       for(int i=0; i<npes; i++){
           int pid = i;
           int from = i * nelements, to = (i+1)*nelements;
           printf("Process no: %d \n Sorted Array: ", pid);
           if(pid == npes - 1) to = n;
           for(int j=from; j<to; j++) printf("%d ", temp[j]);
           printf("\n");
       }
   }

   // Merge Children's arrays
```

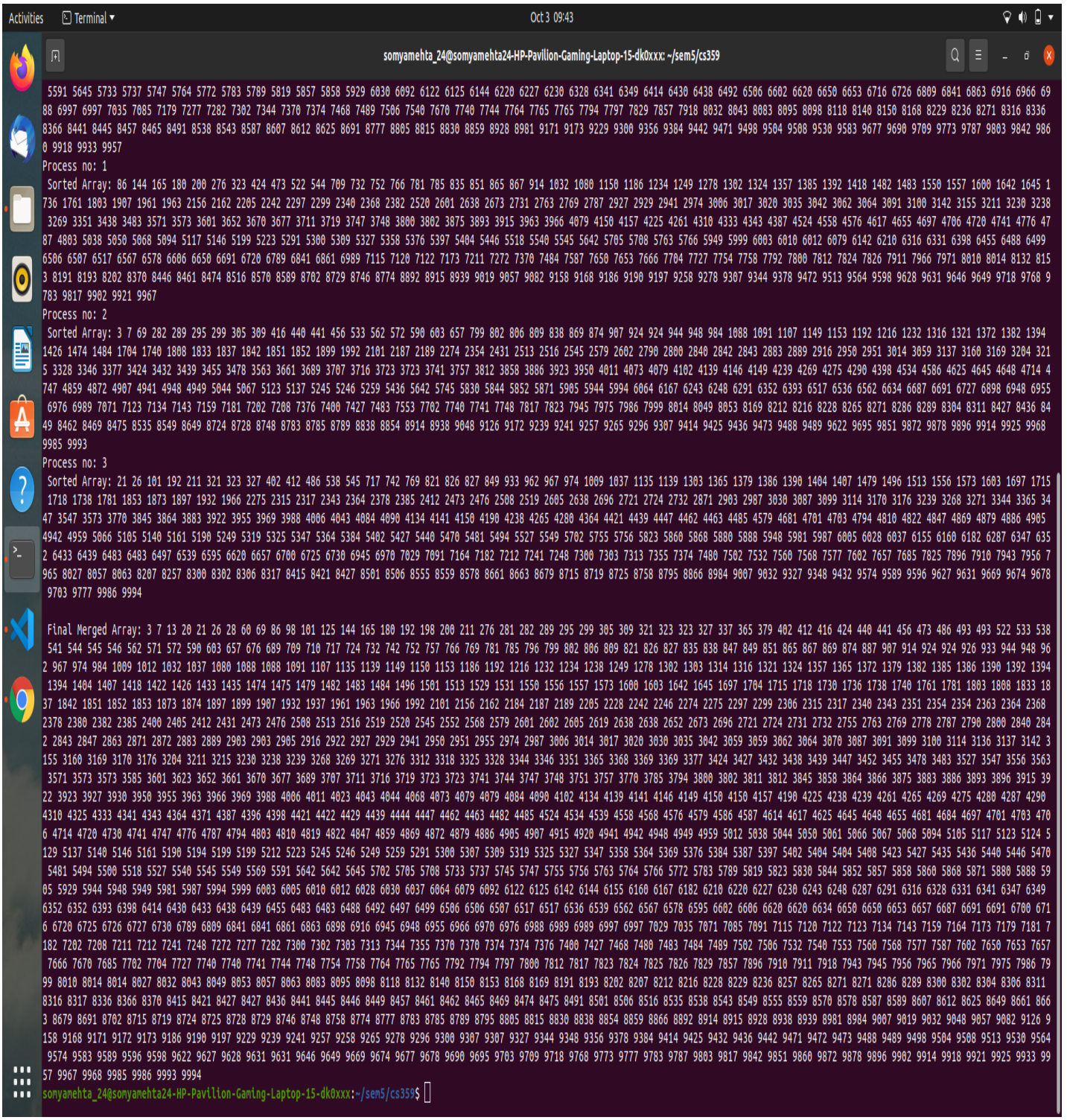
```
   int children1 = 2*myRank + 1, children2 = 2*myRank + 2;
   int parent = (myRank - 1) / 2;
   int childrenSize1 = 0, childrenSize2 = 0;
   int *childrenArray1 = NULL, *childrenArray2 = NULL;


   if(children2 < npes){
       MPI_Recv(&childrenSize2, 1, MPI_INT, children2, 0, MPI_COMM_WORLD,
&status);
       childrenArray2 = malloc(childrenSize2 * sizeof(int));
       MPI_Recv(childrenArray2, childrenSize2, MPI_INT, children2, 52,
MPI_COMM_WORLD, &status);
   }
   if(children1 < npes){
       MPI_Recv(&childrenSize1, 1, MPI_INT, children1, 0, MPI_COMM_WORLD,
&status);
       childrenArray1 = malloc(childrenSize1 * sizeof(int));
       MPI_Recv(childrenArray1, childrenSize1, MPI_INT, children1, 52,
MPI_COMM_WORLD, &status);
   }
   int* finalarr = NULL;
   if(childrenSize1 + childrenSize2 > 0){
       finalarr = malloc((childrenSize1 + childrenSize2) * sizeof(int));
       finalarr = merge(finalarr, childrenArray1, childrenArray2,
childrenSize1, childrenSize2);
   }


   // Merge the children's array with current process' array
   int *finalArray = malloc((childrenSize1 + childrenSize2 + nelements) *
sizeof(int));
   finalArray = merge(finalArray, finalarr, arr,
childrenSize1+childrenSize2, nelements);
```

```
    // Send final array to parent array
    if(myRank != 0){
        int totalSize = childrenSize1 + childrenSize2 + nelements;
        MPI_Send(&totalSize, 1, MPI_INT, parent, 0, MPI_COMM_WORLD);
        MPI_Send(finalArray, totalSize, MPI_INT, parent, 52,
MPI_COMM_WORLD);
    }

    // stop process 0 until others complete
    MPI_Barrier(MPI_COMM_WORLD);

    // Print the final result in the process  0
    if(myRank == 0){
        printf("\n Final Merged Array: ");
        for(int i=0; i<n; i++) printf("%d ", finalArray[i]);
        printf("\n");
    }

    MPI_Finalize();
    return 0;
}
```

SCREENSHOTS (Given 3 inputs where n=10,100 and 1000)

(Note: n is not divisible by npes(number of processors) then also the case is handled)


Activities
Terminal
Oct 3 09:43
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/sem5/cs359
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/sem5/cs359$ mpicc Assignment2.c -o ass1
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/sem5/cs359$ mpirun -n 4 ./ass1
Enter Size of Array: 10
Process no: 0
Sorted Array: 887 9384
Process no: 1
Sorted Array: 5291 6720
Process no: 2
Sorted Array: 4747 7986
Process no: 3
Sorted Array: 2275 4084 8302 9327
Final Merged Array: 887 2275 4084 4747 5291 6720 7986 8302 9327 9384
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/sem5/cs359$ mpirun -n 4 ./ass1
Enter Size of Array: 100
Process no: 0
Sorted Array: 28 60 493 541 887 1422 2363 2568 2778 3427 3927 5212 5369 5387 5737 5783 6430 6650 6916 7764 7794 8336 8691 9173 9384
Process no: 1
Sorted Array: 180 865 1150 2368 2731 3062 3573 4343 4576 4655 5038 5050 5094 5291 6003 6316 6499 6720 6789 8570 8746 8774 9278 9628 9768
Process no: 2
Sorted Array: 441 603 802 1394 1426 1740 2274 2516 2842 3377 3439 4073 4747 4859 4941 5259 6976 7143 7202 7986 8169 8265 9265 9896 9925
Process no: 3
Sorted Array: 26 538 545 1386 1404 1718 2275 2364 2605 2903 3955 3969 4084 4150 4280 4942 5364 5402 7029 7965 8302 8415 8715 9327 9574
Final Merged Array: 26 28 60 180 441 493 538 541 545 603 802 865 887 1150 1386 1394 1404 1422 1426 1718 1740 2274 2275 2363 2364 2368 2516 2568 2605 2731 2778 2842 2903 3062 3377 3427 3439 3573 3927 3955 3969 4073 4084 4150 4280 4343 4576 4655 4747 4859 4941 4942 5038 5050 5094 5212 5259 5291 5364 5369 5387 5402 5737 5783 6003 6316 6430 6499 6650 6720 6789 6916 6976 7029 7143 7202 7764 7794 7965 7986 8169 8265 8302 8336 8415 8570 8691 8715 8746 8774 9173 9265 9278 9327 9384 9574 9628 9768 9896 9925
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/sem5/cs359$ mpirun -n 4 ./ass1
Enter Size of Array: 1000
Process no: 0
Sorted Array: 13 20 28 60 98 125 198 281 337 365 379 493 493 541 546 571 676 689 710 724 757 796 847 887 926 1012 1088 1238 1314 1394 1422 1433 1435 1475 1501 1529 1531 1730 1874 1937 2184 2228 2246 2306 2351 2354 2363 2380 2400 2405 2552 2568 2619 2652 2755 2778 2847 2863 2872 2903 2905 2922 2955 3059 3070 3136 3276 3312 3318 3325 3368 3369 3369 3427 3452 3527 3556 3585 3623 3744 3751 3785 3794 3811 3866 3896 3927 3930 4023 4044 4068 4287 4325 4341 4371 4396 4422 4429 4444 4482 4539 4568 4587 4614 4684 4730 4819 4915 4920 5012 5061 5124 5129 5194 5199 5212 5307 5369 5387 5404 5408 5423 5435 5500 5569 5591 5645 5733 5737 5747 5764 5772 5783 5789 5819 5857 5858 5929 6030 6092 6122 6125 6144 6220 6227 6230 6328 6341 6349 6414 6430 6438 6492 6506 6602 6620 6650 6653 6716 6726 6809 6841 6863 6916 6966 6988 6997 6997 7035 7085 7179 7277 7282 7302 7344 7370 7374 7468 7489 7506 7540 7670 7740 7744 7764 7765 7765 7794 7797 7829 7857 7918 8032 8043 8083 8095 8098 8118 8140 8150 8168 8229 8236 8271 8316 8336 8366 8441 8445 8457 8465 8491 8538 8543 8587 8607 8612 8625 8691 8777 8805 8815 8830 8859 8928 8981 9171 9173 9229 9300 9356 9384 9442 9471 9498 9504 9508 9530 9583 9677 9690 9709 9773 9787 9803 9842 9860 9918 9933 9957
Process no: 1
Sorted Array: 86 144 165 180 200 276 323 424 473 522 544 709 732 752 766 781 785 835 851 865 867 914 1032 1080 1150 1186 1234 1249 1278 1302 1324 1357 1385 1392 1418 1482 1483 1550 1557 1600 1642 1645 1736 1761 1803 1907 1961 1963 2156 2162 2205 2242 2297 2299 2340 2368 2382 2520 2601 2638 2673 2731 2763 2769 2787 2927 2929 2941 2974 3006 3017 3020 3035 3042 3062 3064 3091 3100 3142 3155 3211 3230 3238 3269 3351 3438 3483 3571 3573 3601 3652 3670 3677 3711 3719 3747 3748 3800 3802 3875 3893 3915 3963 3966 4079 4150 4157 4225 4261 4310 4333 4343 4387 4524 4558 4576 4617 4655 4697 4706 4720 4741 4776 4787 4803 5038 5050 5068 5094 5117 5146 5199 5223 5291 5300 5309 5327 5358 5376 5397 5404 5446 5518 5540 5545 5642 5705 5708 5763 5766 5949 5999 6003 6010 6012 6079 6142 6210 6316 6331 6398 6455 6488 6499 6506 6507 6517 6567 6578 6606 6650 6691 6720 6789 6841 6861 6989 7115 7120 7122 7173 7211 7272 7370 7484 7587 7650 7653 7666 7704 7727 7754 7758 7792 7800 7812 7824 7826 7911 7966 7971 8010 8014 8132 8153 8191 8193 8202 8370 8446 8461 8474 8516 8570 8589 8702 8729 8746 8774 8892 8915 8939 9019 9057 9082 9158 9168 9186 9190 9197 9258 9278 9307 9344 9378 9472 9513 9564 9598 9628 9631 9646 9649 9718 9768 9783 9817 9902 9921 9967
Process no: 2
Sorted Array: 3 7 69 282 289 295 299 305 309 416 440 441 456 533 562 572 590 603 657 799 802 806 809 838 869 874 907 924 924 944 948 984 1088 1091 1107 1149 1153 1192 1216 1232 1316 1321 1372 1382 1394 1426 1474 1484 1704 1740 1808 1833 1837 1842 1851 1852 1899 1992 2101 2187 2189 2274 2354 2431 2513 2516 2545 2579 2602 2790 2800 2840 2842 2843 2883 2889 2916 2950 2951 3014 3059 3137 3160 3169 3204 3215 3328 3346 3377 3424 3432 3439 3455 3478 3563 3661 3689 3707 3716 3723 3723 3741 3757 3812 3858 3886 3923 3950 4011 4073 4079 4102 4139 4146 4149 4239 4269 4275 4290 4398 4534 4586 4625 4645 4648 4714 4747 4859 4872 4907 4941 4948 4949 5044 5067 5123 5137 5245 5246 5259 5436 5642 5745 5830 5844 5852 5871 5905 5944 5994 6064 6167 6243 6248 6291 6352 6393 6517 6536 6562 6634 6687 6691 6727 6898 6948 6955 6976 6989 7071 7123 7134 7143 7159 7181 7202 7208 7376 7400 7427 7483 7553 7702 7740 7741 7748 7817 7823 7945 7975 7986 7999 8014 8049 8053 8169 8212 8216 8228 8265 8271 8286 8289 8304 8311 8427 8436 8449 8462 8469 8475 8535 8549 8649 8724 8728 8748 8783 8785 8789 8838 8854 8914 8938 9048 9126 9172 9239 9241 9257 9265 9296 9307 9414 9425 9436 9473 9488 9489 9622 9695 9851 9872 9878 9896 9914 9925 9968 9985 9993
Process no: 3
Sorted Array: 21 26 101 192 211 321 323 327 402 412 486 538 545 717 742 769 821 826 827 849 933 962 967 974 1009 1037 1135 1139 1303 1365 1379 1386 1390 1404 1407 1479 1496 1513 1556 1573 1603 1697 1715 1718 1738 1781 1853 1873 1897 1932 1966 2275 2315 2317 2343 2364 2378 2385 2412 2473 2476 2508 2519 2605 2638 2696 2721 2724 2732 2871 2903 2987 3030 3087 3099 3114 3170 3176 3239 3268 3271 3344 3365 34


Activities
Terminal
Oct 3 09:43
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/sem5/cs359
5591 5645 5733 5737 5747 5764 5772 5783 5789 5819 5857 5858 5929 6030 6092 6122 6125 6144 6220 6227 6230 6328 6341 6349 6414 6430 6438 6492 6506 6602 6620 6650 6653 6716 6726 6809 6841 6863 6916 6966 6988 6997 6997 7035 7085 7179 7277 7282 7302 7344 7370 7374 7468 7489 7506 7540 7670 7740 7744 7764 7765 7765 7794 7797 7829 7857 7918 8032 8043 8083 8095 8098 8118 8140 8150 8168 8229 8236 8271 8316 8336 8366 8441 8445 8457 8465 8491 8538 8543 8587 8607 8612 8625 8691 8777 8805 8815 8830 8859 8928 8981 9171 9173 9229 9300 9356 9384 9442 9471 9498 9504 9508 9530 9583 9677 9690 9709 9773 9787 9803 9842 9860 9918 9933 9957
Process no: 1
Sorted Array: 86 144 165 180 200 276 323 424 473 522 544 709 732 752 766 781 785 835 851 865 867 914 1032 1080 1150 1186 1234 1249 1278 1302 1324 1357 1385 1392 1418 1482 1483 1550 1557 1600 1642 1645 1736 1761 1803 1907 1961 1963 2156 2162 2205 2242 2297 2299 2340 2368 2382 2520 2601 2638 2673 2731 2763 2769 2787 2927 2929 2941 2974 3006 3017 3020 3035 3042 3062 3064 3091 3100 3142 3155 3211 3230 3238 3269 3351 3438 3483 3571 3573 3601 3652 3670 3677 3711 3719 3747 3748 3800 3802 3875 3893 3915 3963 3966 4079 4150 4157 4225 4261 4310 4333 4343 4387 4524 4558 4576 4617 4655 4697 4706 4720 4741 4776 4787 4803 5038 5050 5068 5094 5117 5146 5199 5223 5291 5300 5309 5327 5358 5376 5397 5404 5446 5518 5540 5545 5642 5705 5708 5763 5766 5949 5999 6003 6010 6012 6079 6142 6210 6316 6331 6398 6455 6488 6499 6506 6507 6517 6567 6578 6606 6650 6691 6720 6789 6841 6861 6989 7115 7120 7122 7173 7211 7272 7370 7484 7587 7650 7653 7666 7704 7727 7754 7758 7792 7800 7812 7824 7826 7911 7966 7971 8010 8014 8132 8153 8191 8193 8202 8370 8446 8461 8474 8516 8570 8589 8702 8729 8746 8774 8892 8915 8939 9019 9057 9082 9158 9168 9186 9190 9197 9258 9278 9307 9344 9378 9472 9513 9564 9598 9628 9631 9646 9649 9718 9768 9783 9817 9902 9921 9967
Process no: 2
Sorted Array: 3 7 69 282 289 295 299 305 309 416 440 441 456 533 562 572 590 603 657 799 802 806 809 838 869 874 907 924 924 944 948 984 1088 1091 1107 1149 1153 1192 1216 1232 1316 1321 1372 1382 1394 1426 1474 1484 1704 1740 1808 1833 1837 1842 1851 1852 1899 1992 2101 2187 2189 2274 2354 2431 2513 2516 2545 2579 2602 2790 2800 2840 2842 2843 2883 2889 2916 2950 2951 3014 3059 3137 3160 3169 3204 3215 3328 3346 3377 3424 3432 3439 3455 3478 3563 3661 3689 3707 3716 3723 3723 3741 3757 3812 3858 3886 3923 3950 4011 4073 4079 4102 4139 4146 4149 4239 4269 4275 4290 4398 4534 4586 4625 4645 4648 4714 4747 4859 4872 4907 4941 4948 4949 5044 5067 5123 5137 5245 5246 5259 5436 5642 5745 5830 5844 5852 5871 5905 5944 5994 6064 6167 6243 6248 6291 6352 6393 6517 6536 6562 6634 6687 6691 6727 6898 6948 6955 6976 6989 7071 7123 7134 7143 7159 7181 7202 7208 7376 7400 7427 7483 7553 7702 7740 7741 7748 7817 7823 7945 7975 7986 7999 8014 8049 8053 8169 8212 8216 8228 8265 8271 8286 8289 8304 8311 8427 8436 8449 8462 8469 8475 8535 8549 8649 8724 8728 8748 8783 8785 8789 8838 8854 8914 8938 9048 9126 9172 9239 9241 9257 9265 9296 9307 9414 9425 9436 9473 9488 9489 9622 9695 9851 9872 9878 9896 9914 9925 9968 9985 9993
Process no: 3
Sorted Array: 21 26 101 192 211 321 323 327 402 412 486 538 545 717 742 769 821 826 827 849 933 962 967 974 1009 1037 1135 1139 1303 1365 1379 1386 1390 1404 1407 1479 1496 1513 1556 1573 1603 1697 1715 1718 1738 1781 1853 1873 1897 1932 1966 2275 2315 2317 2343 2364 2378 2385 2412 2473 2476 2508 2519 2605 2638 2696 2721 2724 2732 2871 2903 2987 3030 3087 3099 3114 3170 3176 3239 3268 3271 3344 3365 3447 3547 3573 3770 3845 3864 3883 3922 3955 3969 3988 4006 4043 4084 4090 4134 4141 4150 4190 4238 4265 4280 4364 4421 4439 4447 4462 4463 4485 4579 4681 4701 4703 4794 4810 4822 4847 4869 4879 4886 4905 4942 4959 5066 5105 5140 5161 5190 5249 5319 5325 5347 5364 5384 5402 5427 5440 5470 5481 5494 5527 5549 5702 5755 5756 5823 5860 5868 5880 5888 5948 5981 5987 6005 6028 6037 6155 6160 6182 6287 6347 6352 6433 6439 6483 6483 6497 6539 6595 6620 6657 6700 6725 6730 6945 6970 7029 7091 7164 7182 7212 7241 7248 7300 7303 7313 7355 7374 7480 7502 7532 7560 7568 7577 7602 7657 7685 7825 7896 7910 7943 7956 7965 8027 8057 8063 8207 8257 8300 8302 8306 8317 8415 8421 8427 8501 8506 8555 8559 8578 8661 8663 8679 8715 8719 8725 8758 8795 8866 8984 9007 9032 9327 9348 9432 9574 9589 9596 9627 9631 9669 9674 9678 9703 9777 9986 9994
Final Merged Array: 3 7 13 20 21 26 28 60 69 86 98 101 125 144 165 180 192 198 200 211 276 281 282 289 295 299 305 309 321 323 323 327 337 365 379 402 412 416 424 440 441 456 473 486 493 493 522 533 538 541 544 545 546 562 571 572 590 603 657 676 689 709 710 717 724 732 742 752 757 766 769 781 785 796 799 802 806 809 821 826 827 835 838 847 849 851 865 867 869 874 887 907 914 924 924 926 933 944 948 962 967 974 984 1009 1012 1032 1037 1080 1088 1088 1091 1107 1135 1139 1149 1150 1153 1186 1192 1216 1232 1234 1238 1249 1278 1302 1303 1314 1316 1321 1324 1357 1365 1372 1379 1382 1385 1386 1390 1392 1394 1394 1404 1407 1418 1422 1426 1433 1435 1474 1475 1479 1482 1483 1484 1496 1501 1513 1529 1531 1550 1556 1557 1573 1600 1603 1642 1645 1697 1704 1715 1718 1730 1736 1738 1740 1761 1781 1803 1808 1833 1837 1842 1851 1852 1853 1873 1874 1897 1899 1907 1932 1937 1961 1963 1966 1992 2101 2156 2162 2184 2187 2189 2205 2228 2242 2246 2274 2275 2297 2299 2306 2315 2317 2340 2343 2351 2354 2354 2363 2364 2368 2378 2380 2382 2385 2400 2405 2412 2431 2473 2476 2508 2513 2516 2519 2520 2545 2552 2568 2579 2601 2602 2605 2619 2638 2638 2652 2673 2696 2721 2724 2731 2732 2755 2763 2769 2778 2787 2790 2800 2840 2842 2843 2847 2863 2871 2872 2883 2889 2903 2903 2905 2916 2922 2927 2929 2941 2950 2951 2955 2974 2987 3006 3014 3017 3020 3030 3035 3042 3059 3059 3062 3064 3070 3087 3091 3099 3100 3114 3136 3137 3142 3155 3160 3169 3170 3176 3204 3211 3215 3230 3238 3239 3268 3269 3271 3276 3312 3318 3325 3328 3344 3346 3351 3365 3368 3369 3369 3377 3424 3427 3432 3438 3439 3447 3452 3455 3478 3483 3527 3547 3556 3563 3571 3573 3573 3585 3601 3623 3652 3661 3670 3677 3689 3707 3711 3716 3719 3723 3723 3741 3744 3747 3748 3751 3757 3770 3785 3794 3800 3802 3811 3812 3845 3858 3864 3866 3875 3883 3886 3893 3896 3915 3922 3923 3927 3930 3950 3955 3963 3966 3969 3988 4006 4011 4023 4043 4044 4068 4073 4079 4079 4084 4090 4102 4134 4139 4141 4146 4149 4150 4150 4157 4190 4225 4238 4239 4261 4265 4269 4275 4280 4287 4290 4310 4325 4333 4341 4343 4364 4371 4387 4396 4398 4421 4422 4429 4439 4444 4447 4462 4463 4482 4485 4524 4534 4539 4558 4568 4576 4579 4586 4587 4614 4617 4625 4645 4648 4655 4681 4684 4697 4701 4703 4706 4714 4720 4730 4741 4747 4776 4787 4794 4803 4810 4819 4822 4847 4859 4869 4872 4879 4886 4905 4907 4915 4920 4941 4942 4948 4949 4959 5012 5038 5044 5050 5061 5066 5067 5068 5094 5105 5117 5123 5124 5129 5137 5140 5146 5161 5190 5194 5199 5199 5212 5223 5245 5246 5249 5259 5291 5300 5307 5309 5319 5325 5327 5347 5358 5364 5369 5376 5384 5387 5397 5402 5404 5404 5408 5423 5427 5435 5436 5440 5446 5470 5481 5494 5500 5518 5527 5540 5545 5549 5569 5591 5642 5642 5645 5702 5705 5708 5733 5737 5745 5747 5755 5756 5763 5764 5766 5772 5783 5789 5819 5823 5830 5844 5852 5857 5858 5860 5868 5871 5880 5888 5905 5929 5944 5948 5949 5981 5987 5994 5999 6003 6005 6010 6012 6028 6030 6037 6064 6079 6092 6122 6125 6142 6144 6155 6160 6167 6182 6210 6220 6227 6230 6243 6248 6287 6291 6316 6328 6331 6341 6347 6349 6352 6352 6393 6398 6414 6430 6433 6438 6439 6455 6483 6483 6488 6492 6497 6499 6506 6506 6507 6517 6517 6536 6539 6562 6567 6578 6595 6602 6606 6620 6620 6634 6650 6650 6653 6657 6687 6691 6691 6700 6716 6720 6725 6726 6727 6730 6789 6809 6841 6841 6861 6863 6898 6916 6945 6948 6955 6966 6970 6976 6988 6989 6989 6997 6997 7029 7035 7071 7085 7091 7115 7120 7122 7123 7134 7143 7159 7164 7173 7179 7181 7182 7202 7208 7211 7212 7241 7248 7272 7277 7282 7300 7302 7303 7313 7344 7355 7370 7370 7374 7374 7376 7400 7427 7468 7480 7483 7484 7489 7502 7506 7532 7540 7553 7560 7568 7577 7587 7602 7650 7653 7657 7666 7670 7685 7702 7704 7727 7740 7740 7741 7744 7748 7754 7758 7764 7765 7765 7792 7794 7797 7800 7812 7817 7823 7824 7825 7826 7829 7857 7896 7910 7911 7918 7943 7945 7956 7965 7966 7971 7975 7986 7999 8010 8014 8014 8027 8032 8043 8049 8053 8057 8063 8083 8095 8098 8118 8132 8140 8150 8153 8168 8169 8191 8193 8202 8207 8212 8216 8228 8229 8236 8257 8265 8271 8271 8286 8289 8300 8302 8304 8306 8311 8316 8317 8336 8366 8370 8415 8421 8427 8427 8436 8441 8445 8446 8449 8457 8461 8462 8465 8469 8474 8475 8491 8501 8506 8516 8535 8538 8543 8549 8555 8559 8570 8578 8587 8589 8607 8612 8625 8649 8661 8663 8679 8691 8702 8715 8719 8724 8725 8728 8729 8746 8748 8758 8774 8777 8783 8785 8789 8795 8805 8815 8830 8838 8854 8859 8866 8892 8914 8915 8928 8938 8939 8981 8984 9007 9019 9032 9048 9057 9082 9126 9158 9168 9171 9172 9173 9186 9190 9197 9229 9239 9241 9257 9258 9265 9278 9296 9300 9307 9307 9327 9344 9348 9356 9378 9384 9414 9425 9432 9436 9442 9471 9472 9473 9488 9489 9498 9504 9508 9513 9530 9564 9574 9583 9589 9596 9598 9622 9627 9628 9631 9631 9646 9649 9669 9674 9677 9678 9690 9695 9703 9709 9718 9768 9773 9777 9783 9787 9803 9817 9842 9851 9860 9872 9878 9896 9902 9914 9918 9921 9925 9933 9957 9967 9968 9985 9986 9993 9994
somyamehta_24@somyamehta24-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/sem5/cs359$