Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Class: Final Year (Computer Science and Engineering)

Year: 2022-23 **Semester:** 7

Course: High Performance Computing Lab

Practical No. 03

Exam Seat No:

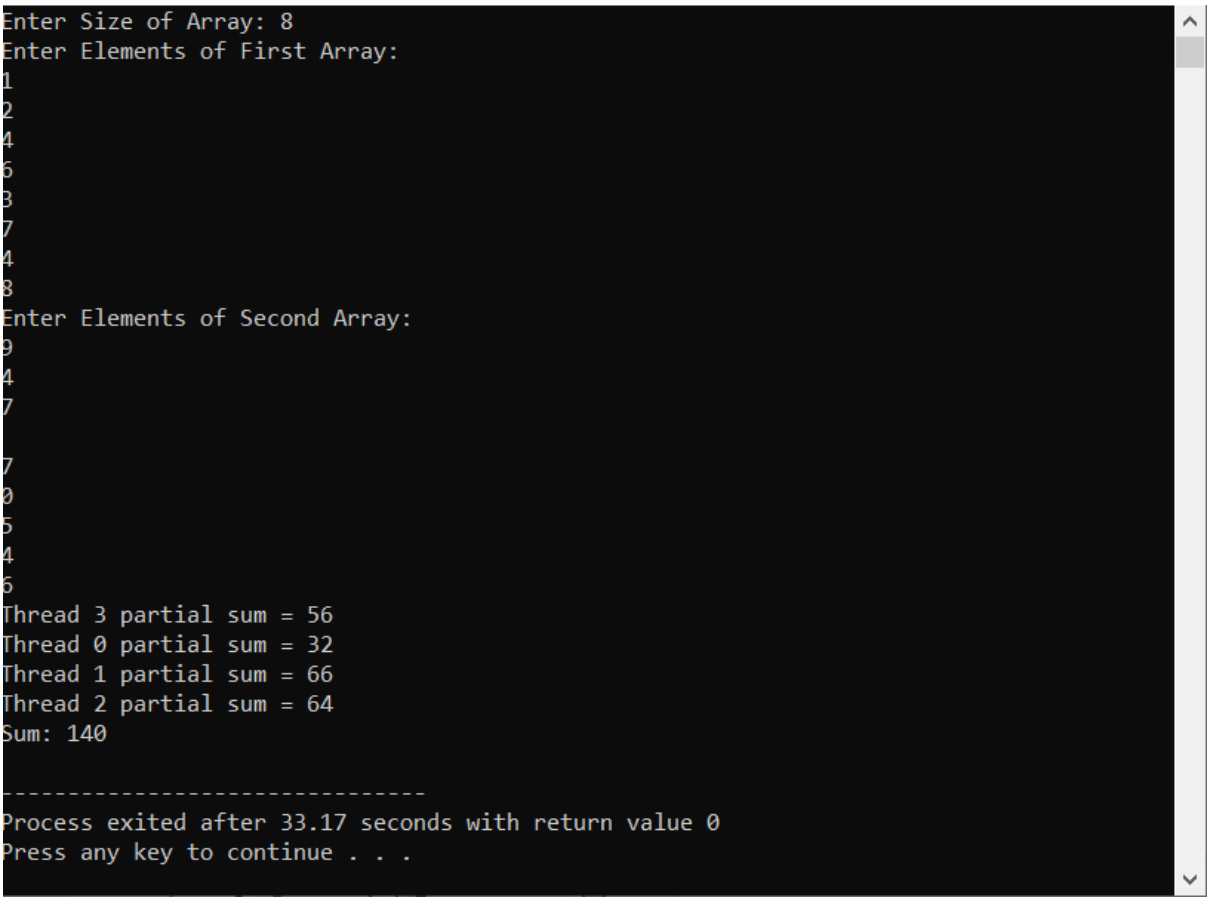
1. 2019BTECS00033 --- Teknath K jha

Title of practical:

Solve give questions

1. **Problem Statement 1:** Analyse and implement a Parallel code for below program using openMP

Screenshot 1:

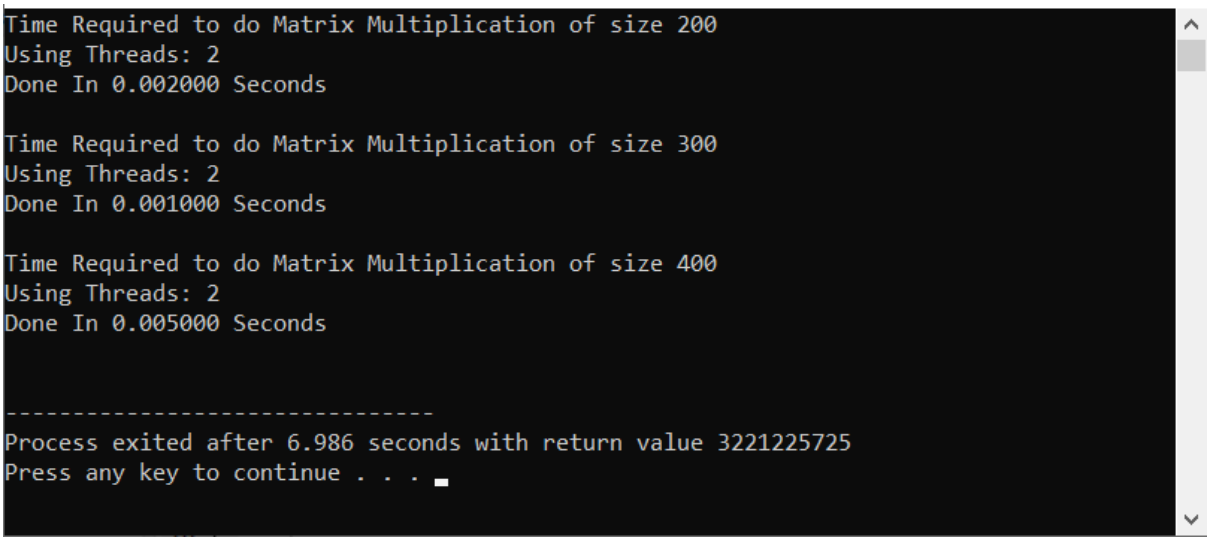


Information 1:
Here I have to modify the give code for finding minimum from array in which I have used openmp clauses like schedule , static , private , for , reduction
I have printed intermediate steps on console .
Code : On github

Problem Statement 3:

Problem Statement 2:

2. Write OpenMP code for two 2D Matrix addition, vary the size of your matrices from 250, 500, 750, 1000, and 2000 and measure the runtime with one thread (Use functions in C in calculate the execution time or use GPROF)
    i. For each matrix size, change the number of threads from 2,4,8., and plot the speedup versus the number of threads.
    ii. Explain whether or not the scaling behaviour is as expected.



```
Time Required to do Matrix Multiplication of size 200
Using Threads: 4
Done In 0.003000 Seconds

Time Required to do Matrix Multiplication of size 300
Using Threads: 4
Done In 0.001000 Seconds

Time Required to do Matrix Multiplication of size 400
Using Threads: 4
Done In 0.004000 Seconds


--------------------------------
Process exited after 6.82 seconds with return value 3221225725
Press any key to continue . . .
```

Problem Statement 3: For 1D Vector (size=200) and scalar addition, Write a OpenMP code with the following:

i. **Use STATIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**

For array size 100000 with 4 threads :

i. **Use DYNAMIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**

For n=10000 with 4 threads

```
Thread number 2, executing iteration 2482 first time
Thread number 2, executing iteration 2483 first time
Thread number 2, executing iteration 2484 first time
Thread number 2, executing iteration 2485 first time
Thread number 2, executing iteration 2486 first time
Thread number 2, executing iteration 2487 first time
Thread number 2, executing iteration 2488 first time
Thread number 2, executing iteration 2489 first time
Thread number 2, executing iteration 2490 first time
Thread number 2, executing iteration 2491 first time
Thread number 2, executing iteration 2492 first time
Thread number 2, executing iteration 2493 first time
Thread number 2, executing iteration 2494 first time
Thread number 2, executing iteration 2495 first time
Thread number 2, executing iteration 2496 first time
Thread number 2, executing iteration 2497 first time
Thread number 2, executing iteration 2498 first time
Thread number 2, executing iteration 2499 first timeTime taken : 3.743000


--------------------------------
Process exited after 3.783 seconds with return value 0
Press any key to continue . . .
```

For N= 100000 with 4 threads

```
Thread number 3, executing iteration 99983 first time
Thread number 3, executing iteration 99984 first time
Thread number 3, executing iteration 99985 first time
Thread number 3, executing iteration 99986 first time
Thread number 3, executing iteration 99987 first time
Thread number 3, executing iteration 99988 first time
Thread number 3, executing iteration 99989 first time
Thread number 3, executing iteration 99990 first time
Thread number 3, executing iteration 99991 first time
Thread number 3, executing iteration 99992 first time
Thread number 3, executing iteration 99993 first time
Thread number 3, executing iteration 99994 first time
Thread number 3, executing iteration 99995 first time
Thread number 3, executing iteration 99996 first time
Thread number 3, executing iteration 99997 first time
Thread number 3, executing iteration 99998 first time
Thread number 3, executing iteration 99999 first timeTime taken : 6.180000


--------------------------------
Process exited after 6.218 seconds with return value 0
Press any key to continue . . .
```

i. Demonstrate the use of nowait clause

```
Done In 0.000000 Seconds


Array 1:
        41      67      34      0       69      24      78      58      62      64      5       45      81      27
 61     91      95      42      27      36      91      4       2       53      92      82      21      16      18
 95     47      26      71      38      69      12      67      99      35      94      3       11      22      33
 73     64      41      11      53      68      47      44      62      57      37      59      23      41      29
 78     16      35      90      42      88      6       40      42      64      48      46      5       90      29
 70     50      6       1       93      48      29      23      84      54      56      40      66      76      31
 8      44      39      26      23      37      38      18      82      29      41      33      15      39      58
 4      30      77      6       73      86      21      45      24      72      70      29      77      73      97
 12     86      90      61      36      55      67      55      74      31      52      50      50      41      24
 66     30      7       91      7       37      57      87      53      83      45      9       9       58      21
 88     22      46      6       30      13      68      0       91      62      55      10      59      24      37
 48     83      95      41      2       50      91      36      74      20      96      21      48      99      68
 84     81      34      53      99      18      38      0       88      27      67      28      93      48      83
 7      21      10      17      13      14
```

```
Answer:
        140     166     133     99      168     123     177     157     161     163     104     144     180     126
 160    190     194     141     126     135     190     103     101     152     191     181     120     115     117
 194    146     125     170     137     168     111     166     198     134     193     102     110     121     132
 172    163     140     110     152     167     146     143     161     156     136     158     122     140     128
 177    115     134     189     141     187     105     139     141     163     147     145     104     189     128
 169    149     105     100     192     147     128     122     183     153     155     139     165     175     130
 107    143     138     125     122     136     137     117     181     128     140     132     114     138     157
 103    129     176     105     172     185     120     144     123     171     169     128     176     172     196
 111    185     189     160     135     154     166     154     173     130     151     149     149     140     123
 165    129     106     190     106     136     156     186     152     182     144     108     108     157     120
 187    121     145     105     129     112     167     99      190     161     154     109     158     123     136
 147    182     194     140     101     149     190     135     173     119     195     120     147     198     167
 183    180     133     152     198     117     137     99      187     126     166     127     192     147     182
 106    120     109     116     112     113
--------------------------------
Process exited after 3.023 seconds with return value 0
Press any key to continue . . .
```

Here I have used two threads which will execute with synchroniznation with nowait but as this is vector addition with scalar so independent execution have communication requirement thus nowait will execute .

Code : On github

Github Link:

https://github.com/Teknath-jha/HPC-LAB-2019BTECS00033/tree/main/Assignment-3