Final Year B. Tech., Sem VII 2022-23

High Performance Computing Lab

PRN/ Roll No: 2020BTECS00206

Full Name: SAYALI YOGESH DESAI

Batch: B4

Assignment No. 2

Que 1. What is SPMD?

i. SPMD stands for Single Program Multiple Data.
ii. It is a parallel programming style.
iii. In SPMD style, tasks are split up and run simultaneously on multiple processors with different data.
iv. This is done to achieve results faster.

Que 2. What is Worksharing?

i. A worksharing construct distributes the execution of the corresponding region among the members of the team that encounters it.
ii. Threads execute portions of the region in the context of the implicit tasks that each one is executing.
iii. If the team consists of only one thread then the worksharing region is not executed in parallel.
iv. A worksharing region has no barrier on entry; however, an implied barrier exists at the end of the worksharing region, unless a nowait clause is specified.
v. If a nowait clause is present, an implementation may omit the barrier at the end of the worksharing region.
vi. In this case, threads that finish early may proceed straight to the instructions that follow the worksharing region without waiting for the other members of the team to finish the worksharing region, and without performing a flush operation.
vii. The OpenMP API defines the worksharing constructs:

    a. Sections Construct
    b. Single Constructs
    c. Workshare constructs

Que 3. Implement a program for Vector – Vector Addition

***Parallel-**

- **Code:**

```
#include <omp.h>
#include <stdio.h>
#include <pthread.h>
int main()
{
   int N = 1000;
   int A[1000];
   for(int i=0;i<N;i++)A[i] = i + 1;
   int B[1000];
   for(int i=0;i<N;i++)B[i] = N - i;
   int C[1000] = {0};
   double itime, ftime, exec_time;
   itime = omp_get_wtime();
   #pragma omp parallel for reduction(+ : C)
   for (int i = 0; i < N; i++)
   {
      C[i] = A[i] + B[i];
      printf("Thread: %d Index: %d\n", omp_get_thread_num(),i);
   }
   for(int i=0;i<N;i++)
   {
      printf("%d ", C[i]);
   }
   ftime = omp_get_wtime();
```

```
    exec_time = ftime - itime;
    printf("\nTime taken is %f\n", exec_time);
    printf("\n");
}
```

- **Output:**

```
Thread: 0 Index: 235
Thread: 0 Index: 236
Thread: 0 Index: 237
Thread: 0 Index: 238
Thread: 0 Index: 239
Thread: 0 Index: 240
Thread: 0 Index: 241
Thread: 0 Index: 242
Thread: 0 Index: 243
Thread: 0 Index: 244
Thread: 0 Index: 245
Thread: 0 Index: 246
Thread: 0 Index: 247
Thread: 0 Index: 248
Thread: 0 Index: 249
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001
Time taken is 0.546000
```

***Sequential-**

- **Code:**

```
#include <omp.h>
#include <stdio.h>
#include <pthread.h>
int main()
{
    int N = 1000;
    int A[1000];
    for(int i=0;i<N;i++)A[i] = i + 1;
    int B[1000];
    for(int i=0;i<N;i++)B[i] = N - i;
```

```
    int C[1000] = {0};

    double itime, ftime, exec_time;

    itime = omp_get_wtime();

    for (int i = 0; i < N; i++)

    {

        C[i] = A[i] + B[i];

        printf("Thread: %d Index: %d\n", omp_get_thread_num(),i);

    }

    for(int i=0;i<N;i++)

    {

        printf("%d ", C[i]);

    }

    ftime = omp_get_wtime();

    exec_time = ftime - itime;

    printf("\nTime taken is %f\n", exec_time);

    printf("\n");

}
```

- **Output:**

```
Thread: 0 Index: 993
Thread: 0 Index: 994
Thread: 0 Index: 995
Thread: 0 Index: 996
Thread: 0 Index: 997
Thread: 0 Index: 998
Thread: 0 Index: 999
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 10
01 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1
001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 100
1 1001 1001 1001 1001 1001 1001 1001 1001
Time taken is 0.520000

PS D:\Walchand\7 Semester\HPC\Assignment_2>
```

Que 3. Implement a program for Vector – Scalar Multiplication

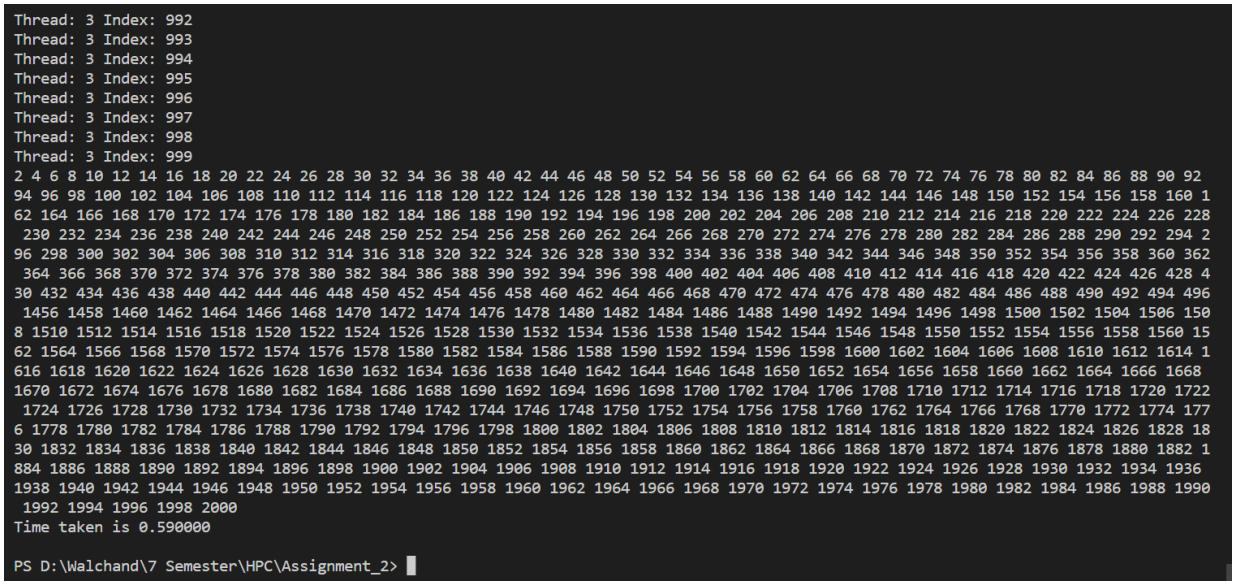
***Parallel-**

- **Code:**

```
#include <omp.h>
#include <stdio.h>
#include <pthread.h>
int main()
{
   int N = 1000;
   int A[1000];
   for(int i=0;i<N;i++)A[i] = i + 1;
   int S = 2;
   double itime, ftime, exec_time;
   itime = omp_get_wtime();
   #pragma omp parallel for
   for (int i = 0; i < N; i++)
   {
      A[i] *= S;
      printf("Thread: %d Index: %d\n", omp_get_thread_num(),i);
   }
   for(int i=0;i<N;i++)
   {
      printf("%d ", A[i]);
   }
   ftime = omp_get_wtime();
   exec_time = ftime - itime;
   printf("\nTime taken is %f\n", exec_time);
```

```
    printf("\n");
}
```

- **Output:**



```
Thread: 3 Index: 992
Thread: 3 Index: 993
Thread: 3 Index: 994
Thread: 3 Index: 995
Thread: 3 Index: 996
Thread: 3 Index: 997
Thread: 3 Index: 998
Thread: 3 Index: 999
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92
94 96 98 100 102 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158 160 1
62 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198 200 202 204 206 208 210 212 214 216 218 220 222 224 226 228
 230 232 234 236 238 240 242 244 246 248 250 252 254 256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 294 2
96 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362
 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406 408 410 412 414 416 418 420 422 424 426 428 4
30 432 434 436 438 440 442 444 446 448 450 452 454 456 458 460 462 464 466 468 470 472 474 476 478 480 482 484 486 488 490 492 494 496
 1456 1458 1460 1462 1464 1466 1468 1470 1472 1474 1476 1478 1480 1482 1484 1486 1488 1490 1492 1494 1496 1498 1500 1502 1504 1506 150
8 1510 1512 1514 1516 1518 1520 1522 1524 1526 1528 1530 1532 1534 1536 1538 1540 1542 1544 1546 1548 1550 1552 1554 1556 1558 1560 15
62 1564 1566 1568 1570 1572 1574 1576 1578 1580 1582 1584 1586 1588 1590 1592 1594 1596 1598 1600 1602 1604 1606 1608 1610 1612 1614 1
616 1618 1620 1622 1624 1626 1628 1630 1632 1634 1636 1638 1640 1642 1644 1646 1648 1650 1652 1654 1656 1658 1660 1662 1664 1666 1668
1670 1672 1674 1676 1678 1680 1682 1684 1686 1688 1690 1692 1694 1696 1698 1700 1702 1704 1706 1708 1710 1712 1714 1716 1718 1720 1722
 1724 1726 1728 1730 1732 1734 1736 1738 1740 1742 1744 1746 1748 1750 1752 1754 1756 1758 1760 1762 1764 1766 1768 1770 1772 1774 177
6 1778 1780 1782 1784 1786 1788 1790 1792 1794 1796 1798 1800 1802 1804 1806 1808 1810 1812 1814 1816 1818 1820 1822 1824 1826 1828 18
30 1832 1834 1836 1838 1840 1842 1844 1846 1848 1850 1852 1854 1856 1858 1860 1862 1864 1866 1868 1870 1872 1874 1876 1878 1880 1882 1
884 1886 1888 1890 1892 1894 1896 1898 1900 1902 1904 1906 1908 1910 1912 1914 1916 1918 1920 1922 1924 1926 1928 1930 1932 1934 1936
1938 1940 1942 1944 1946 1948 1950 1952 1954 1956 1958 1960 1962 1964 1966 1968 1970 1972 1974 1976 1978 1980 1982 1984 1986 1988 1990
 1992 1994 1996 1998 2000
Time taken is 0.590000

PS D:\Walchand\7 Semester\HPC\Assignment_2>
```

***Sequential-**

- **Code:**

```
#include <omp.h>
#include <stdio.h>
#include <pthread.h>
int main()
{
    int N = 1000;
    int A[1000];
    for(int i=0;i<N;i++)A[i] = i + 1;
    int S = 2;
    double itime, ftime, exec_time;
    itime = omp_get_wtime();
    for (int i = 0; i < N; i++)
```

```
    {
        A[i] *= S;
        printf("Thread: %d Index: %d\n", omp_get_thread_num(),i);
    }
    for(int i=0;i<N;i++)
    {
        printf("%d ", A[i]);
    }
    ftime = omp_get_wtime();
    exec_time = ftime - itime;
    printf("\nTime taken is %f\n", exec_time);
    printf("\n");
}
```

- **Output:**

```
Thread: 0 Index: 988
Thread: 0 Index: 989
Thread: 0 Index: 990
Thread: 0 Index: 991
Thread: 0 Index: 992
Thread: 0 Index: 993
Thread: 0 Index: 994
Thread: 0 Index: 995
Thread: 0 Index: 996
Thread: 0 Index: 997
Thread: 0 Index: 998
Thread: 0 Index: 999
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92
94 96 98 100 102 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158 160 1
62 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198 200 202 204 206 208 210 212 214 216 218 220 222 224 226 228
 230 232 234 236 238 240 242 244 246 248 250 252 254 256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 294 2
96 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362
 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406 408 410 412 414 416 418 420 422 424 426 428 4
30 432 434 436 438 440 442 444 446 448 450 452 454 456 458 460 462 464 466 468 470 472 474 476 478 480 482 484 486 488 490 492 494 496
 498 500 502 504 506 508 510 512 514 516 518 520 522 524 526 528 530 532 534 536 538 540 542 544 546 548 550 552 554 556 558 560 562 5
64 566 568 570 572 574 576 578 580 582 584 586 588 590 592 594 596 598 600 602 604 606 608 610 612 614 616 618 620 622 624 626 628 630
884 1886 1888 1890 1892 1894 1896 1898 1900 1902 1904 1906 1908 1910 1912 1914 1916 1918 1920 1922 1924 1926 1928 1930 1932 1934 1936
1938 1940 1942 1944 1946 1948 1950 1952 1954 1956 1958 1960 1962 1964 1966 1968 1970 1972 1974 1976 1978 1980 1982 1984 1986 1988 1990
 1992 1994 1996 1998 2000
Time taken is 0.538000

PS D:\Walchand\7 Semester\HPC\Assignment_2>
```