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

Class: Final Year (Computer Science and Engineering)

Year: 2021-22 **Semester:** 1

Course: High Performance Computing Lab

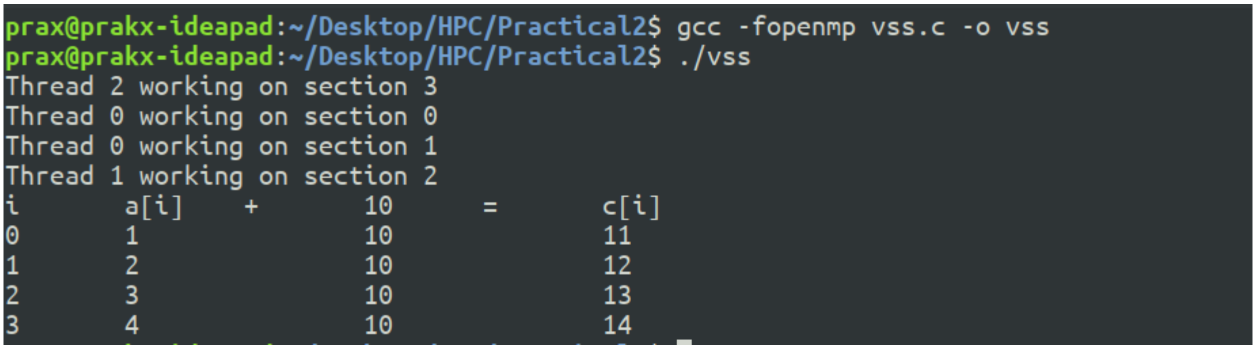
Practical No. 2

Exam Seat No:

1. Exam Seat Number – 2018BTECS00100

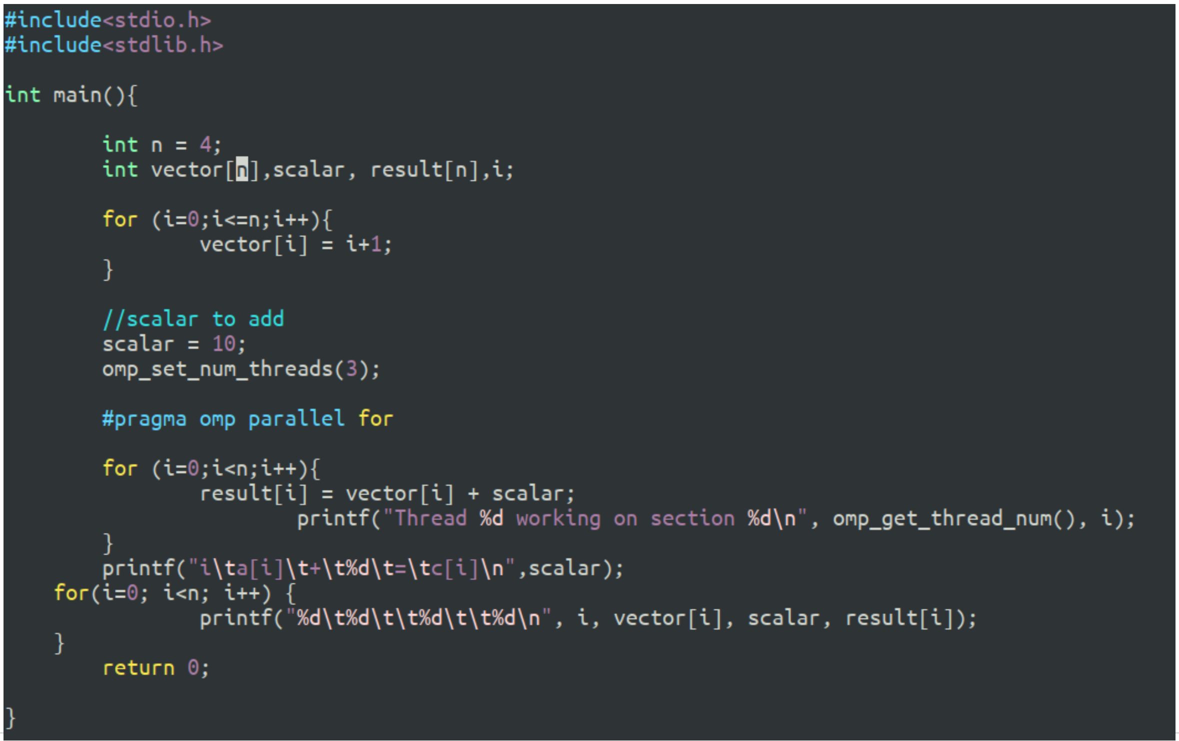
Problem Statement 1:

Screenshot #:



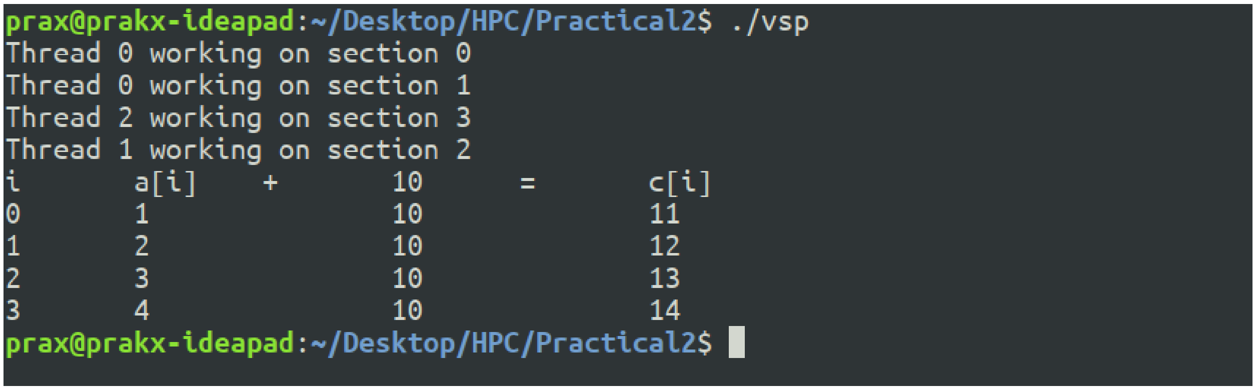
```
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ gcc -fopenmp vss.c -o vss
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ ./vss
Thread 2 working on section 3
Thread 0 working on section 0
Thread 0 working on section 1
Thread 1 working on section 2
i       a[i]    +       10      =       c[i]
0       1               10              11
1       2               10              12
2       3               10              13
3       4               10              14
```

Information #:



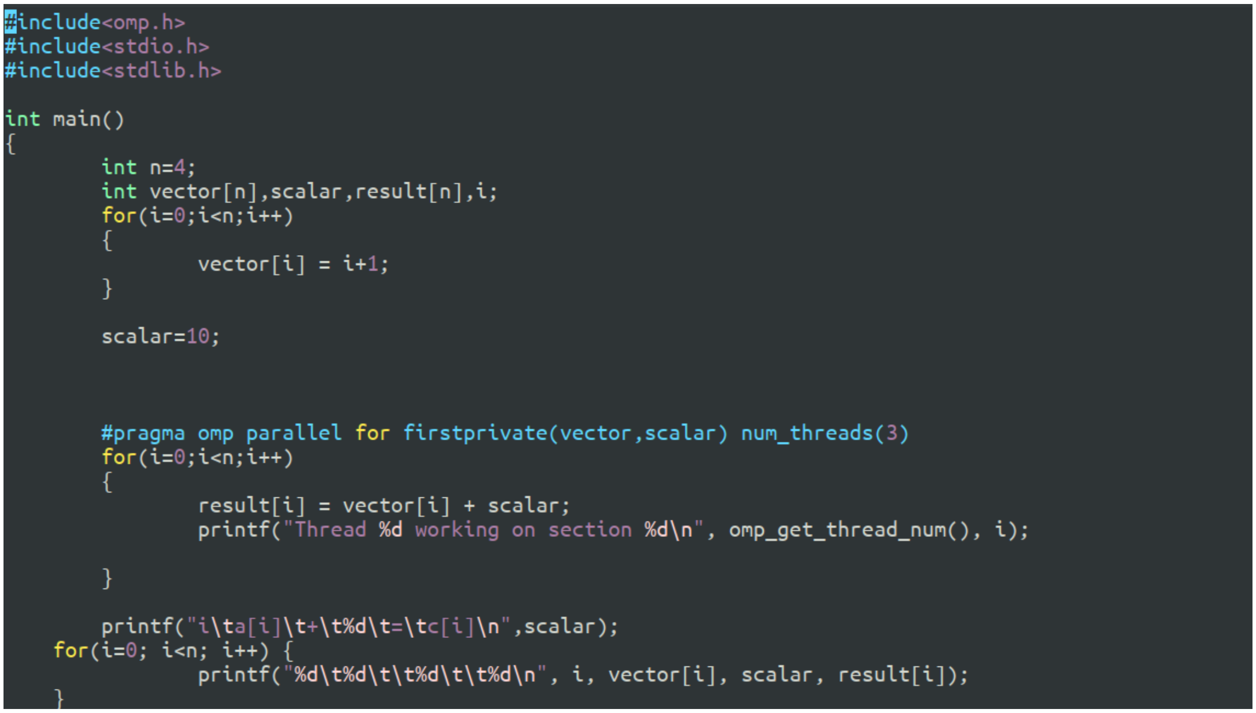
```
#include<stdio.h>
#include<stdlib.h>

int main(){

        int n = 4;
        int vector[n],scalar, result[n],i;

        for (i=0;i<=n;i++){
                vector[i] = i+1;
        }

        //scalar to add
        scalar = 10;
        omp_set_num_threads(3);

        #pragma omp parallel for

        for (i=0;i<n;i++){
                result[i] = vector[i] + scalar;
                        printf("Thread %d working on section %d\n", omp_get_thread_num(), i);
        }
        printf("i\ta[i]\t+\t%d\t=\tc[i]\n",scalar);
    for(i=0; i<n; i++) {
                printf("%d\t%d\t\t%d\t\t%d\n", i, vector[i], scalar, result[i]);
    }
        return 0;

}
```

Screenshot #:



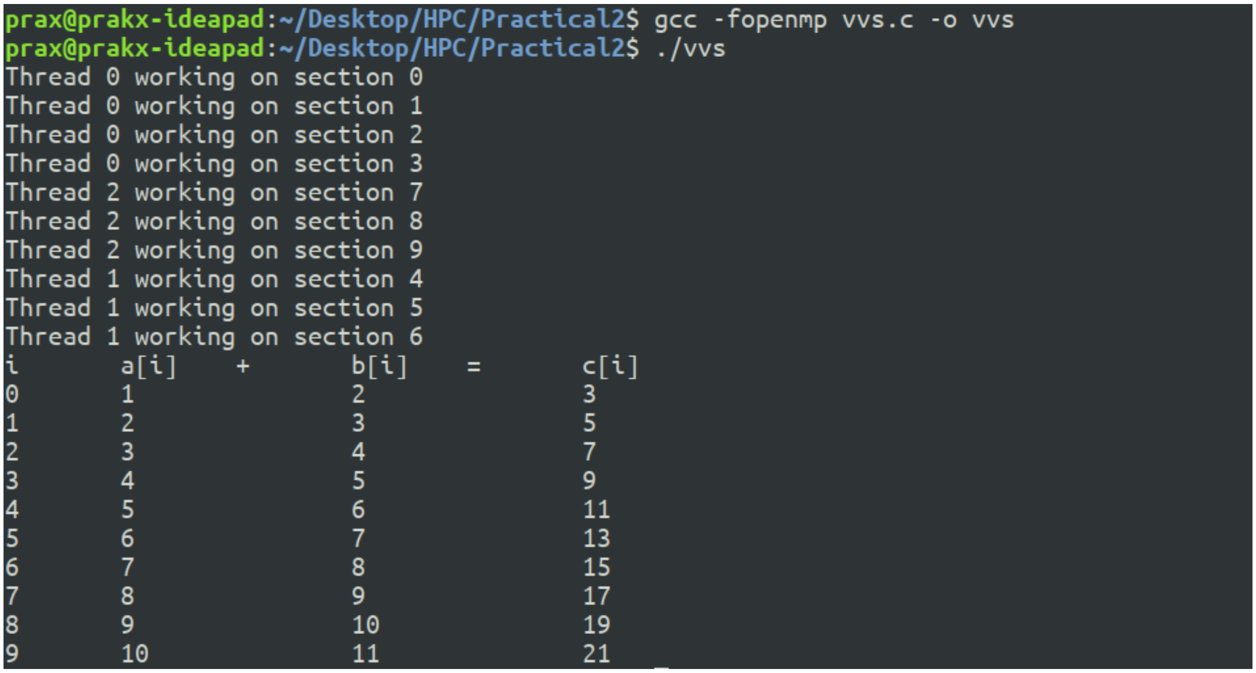
```
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ ./vsp
Thread 0 working on section 0
Thread 0 working on section 1
Thread 2 working on section 3
Thread 1 working on section 2
i       a[i]    +       10      =       c[i]
0       1               10              11
1       2               10              12
2       3               10              13
3       4               10              14
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ 
```

Information #:



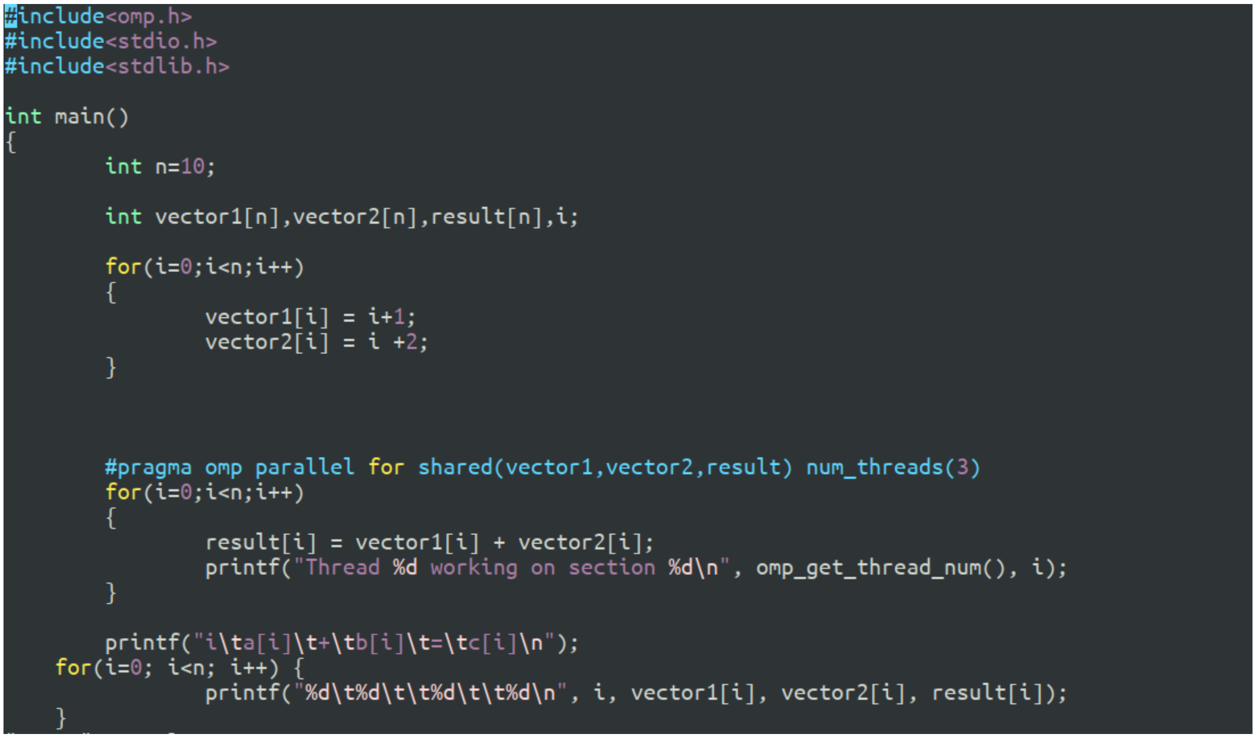
```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>

int main()
{
        int n=4;
        int vector[n],scalar,result[n],i;
        for(i=0;i<n;i++)
        {
                vector[i] = i+1;
        }

        scalar=10;



        #pragma omp parallel for firstprivate(vector,scalar) num_threads(3)
        for(i=0;i<n;i++)
        {
                result[i] = vector[i] + scalar;
                printf("Thread %d working on section %d\n", omp_get_thread_num(), i);

        }

        printf("i\ta[i]\t+\t%d\t=\tc[i]\n",scalar);
    for(i=0; i<n; i++) {
                printf("%d\t%d\t\t%d\t\t%d\n", i, vector[i], scalar, result[i]);
    }
```

Screenshot #:



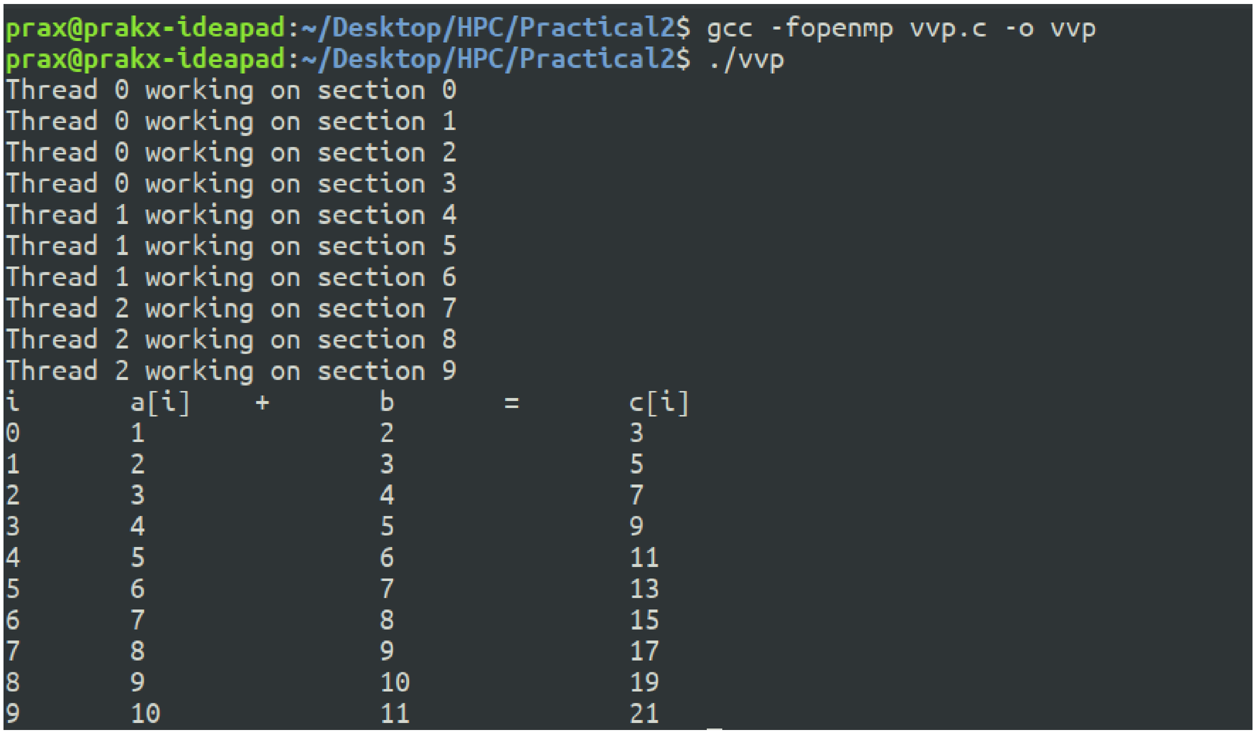
```
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ gcc -fopenmp vvs.c -o vvs
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ ./vvs
Thread 0 working on section 0
Thread 0 working on section 1
Thread 0 working on section 2
Thread 0 working on section 3
Thread 2 working on section 7
Thread 2 working on section 8
Thread 2 working on section 9
Thread 1 working on section 4
Thread 1 working on section 5
Thread 1 working on section 6
i       a[i]    +       b[i]    =       c[i]
0       1               2               3
1       2               3               5
2       3               4               7
3       4               5               9
4       5               6               11
5       6               7               13
6       7               8               15
7       8               9               17
8       9               10              19
9       10              11              21
```

Information #:



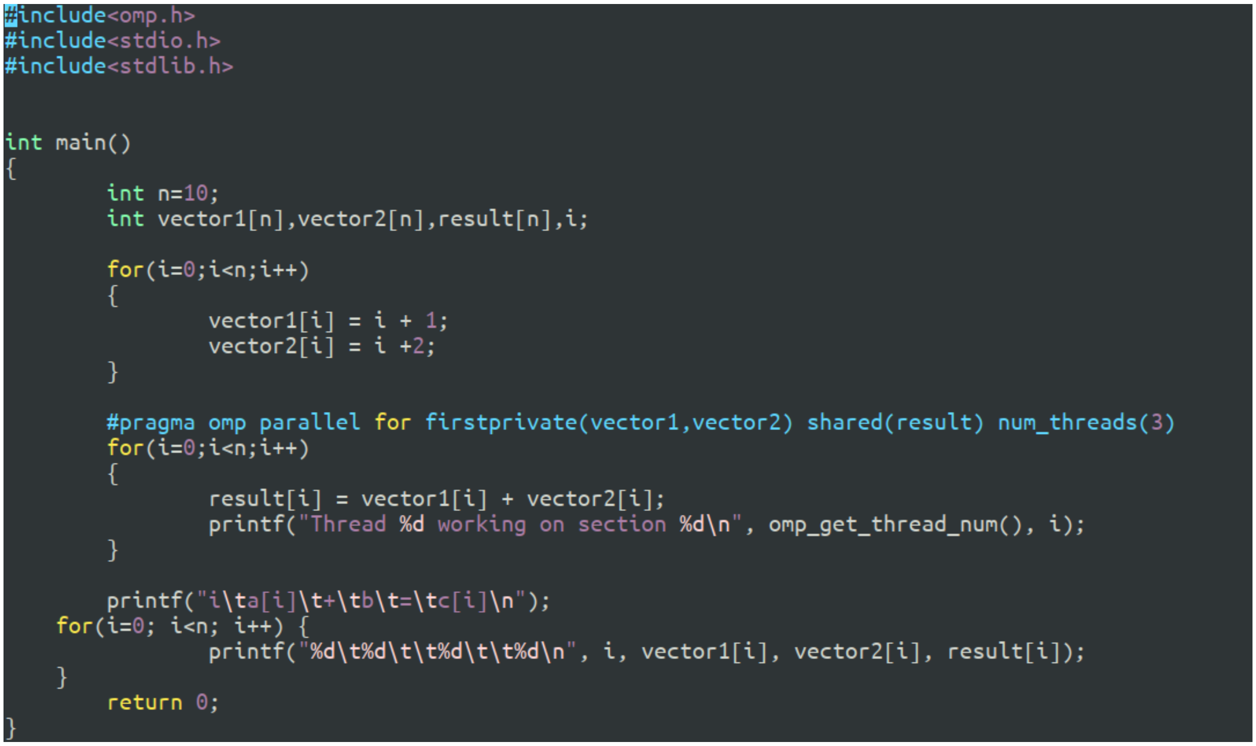
```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>

int main()
{
        int n=10;

        int vector1[n],vector2[n],result[n],i;

        for(i=0;i<n;i++)
        {
                vector1[i] = i+1;
                vector2[i] = i +2;
        }



        #pragma omp parallel for shared(vector1,vector2,result) num_threads(3)
        for(i=0;i<n;i++)
        {
                result[i] = vector1[i] + vector2[i];
                printf("Thread %d working on section %d\n", omp_get_thread_num(), i);
        }

        printf("i\ta[i]\t+\tb[i]\t=\tc[i]\n");
    for(i=0; i<n; i++) {
                printf("%d\t%d\t\t%d\t\t%d\n", i, vector1[i], vector2[i], result[i]);
    }
```

Screenshot #:



```
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ gcc -fopenmp vvp.c -o vvp
prax@prakx-ideapad:~/Desktop/HPC/Practical2$ ./vvp
Thread 0 working on section 0
Thread 0 working on section 1
Thread 0 working on section 2
Thread 0 working on section 3
Thread 1 working on section 4
Thread 1 working on section 5
Thread 1 working on section 6
Thread 2 working on section 7
Thread 2 working on section 8
Thread 2 working on section 9
i       a[i]    +       b       =       c[i]
0       1               2               3
1       2               3               5
2       3               4               7
3       4               5               9
4       5               6               11
5       6               7               13
6       7               8               15
7       8               9               17
8       9               10              19
9       10              11              21
```

Information #:



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


int main()
{
        int n=10;
        int vector1[n],vector2[n],result[n],i;

        for(i=0;i<n;i++)
        {
                vector1[i] = i + 1;
                vector2[i] = i +2;
        }

        #pragma omp parallel for firstprivate(vector1,vector2) shared(result) num_threads(3)
        for(i=0;i<n;i++)
        {
                result[i] = vector1[i] + vector2[i];
                printf("Thread %d working on section %d\n", omp_get_thread_num(), i);
        }

        printf("i\ta[i]\t+\tb\t=\tc[i]\n");
    for(i=0; i<n; i++) {
                printf("%d\t%d\t\t%d\t\t%d\n", i, vector1[i], vector2[i], result[i]);
    }
        return 0;
}
```

1)Setting number of threads to 5
2)Running the hello.c file using openmp in GNU compiler

Github Link: https://github.com/prakx1/HPC-LAB/tree/master/Practical2