NAME:ARYAMAN MISHRA

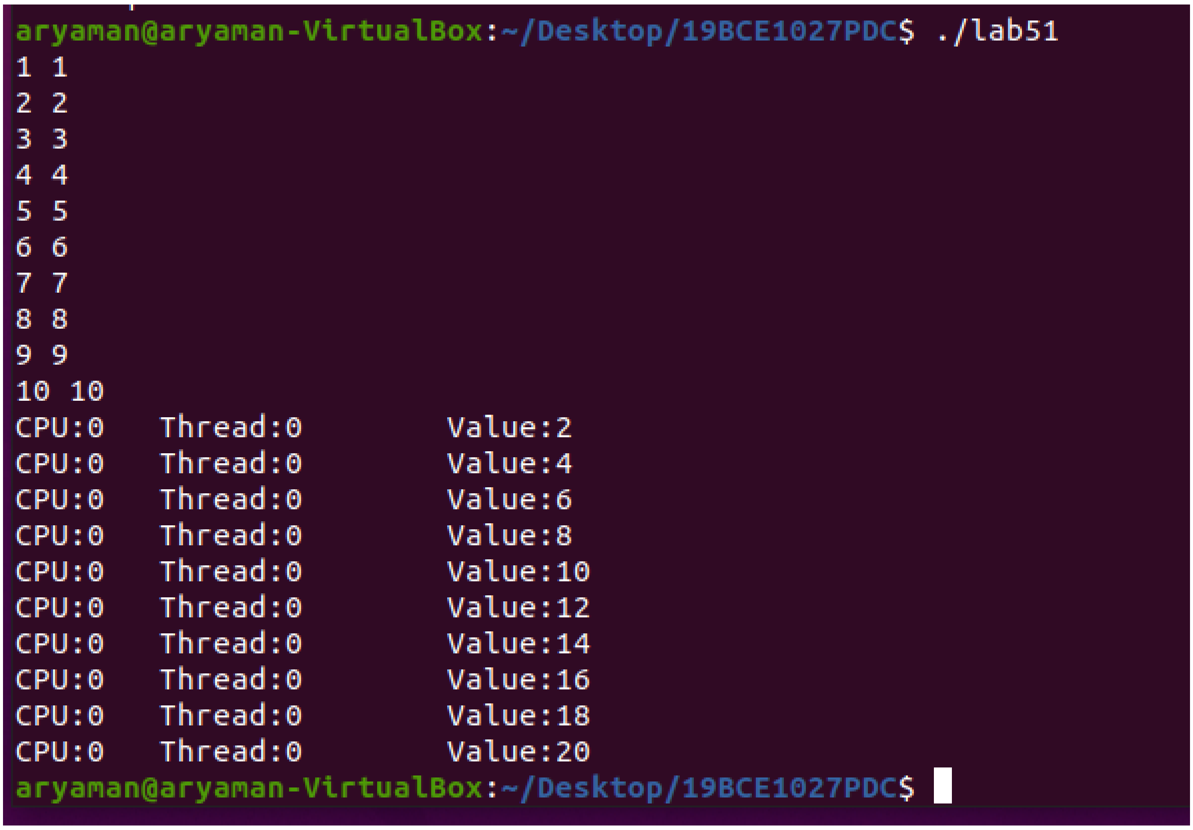
REGISTRATION NUMBER:19BCE1027

Static

Q1 Find the sum of elements of 2 arrays using parallel for.

Code

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

void main()
{
	int a[10],b[10],I,sum[10];
	for(int i=0;i<10;i++){
		scanf("%d",&a[i]);
		scanf("%d",&b[i]);
	}

	#pragma omp parallel for schedule(static,1)
	for(int i=0;i<10;i++){
	sum[i] = a[i]+b[i];
	printf("CPU:%d\tThread:%d\tValue:%d\n",sched_getcpu(),omp_get_thread_num(),s
um[i]);
	}
}
```

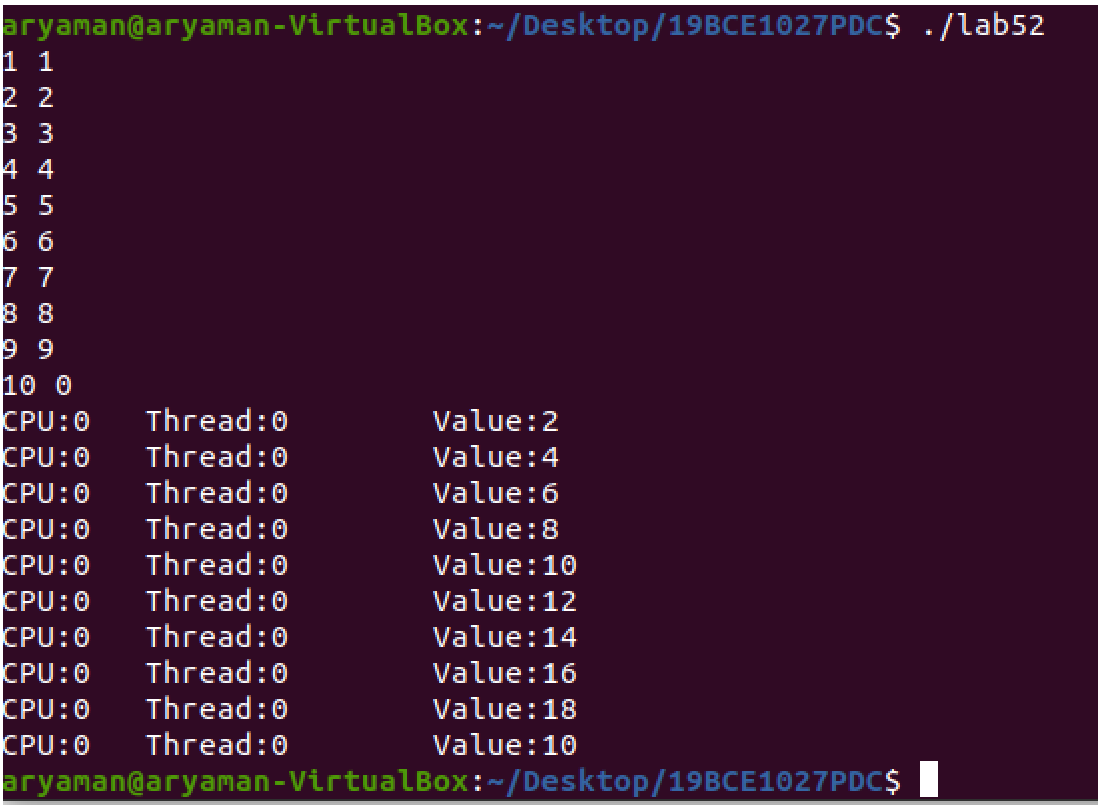


```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab51
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
CPU:0    Thread:0         Value:2
CPU:0    Thread:0         Value:4
CPU:0    Thread:0         Value:6
CPU:0    Thread:0         Value:8
CPU:0    Thread:0         Value:10
CPU:0    Thread:0         Value:12
CPU:0    Thread:0         Value:14
CPU:0    Thread:0         Value:16
CPU:0    Thread:0         Value:18
CPU:0    Thread:0         Value:20
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

Dynamic

```
#include<stdio.h>

#include<omp.h>


void main()

{

        int a[10],b[10],I,sum[10];

        for(int i=0;i<10;i++){

                scanf("%d",&a[i]);

                scanf("%d",&b[i]);

        }


        #pragma omp parallel for schedule(dynamic,1)

        for(int i=0;i<10;i++){

        sum[i] = a[i]+b[i];
```

```
            printf("CPU:%d\tThread:%d\tValue:%d\n",sched_getcpu(),omp_get_thread_num(),s
um[i]);

            }

}
```



```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab52
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 0
CPU:0   Thread:0        Value:2
CPU:0   Thread:0        Value:4
CPU:0   Thread:0        Value:6
CPU:0   Thread:0        Value:8
CPU:0   Thread:0        Value:10
CPU:0   Thread:0        Value:12
CPU:0   Thread:0        Value:14
CPU:0   Thread:0        Value:16
CPU:0   Thread:0        Value:18
CPU:0   Thread:0        Value:10
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

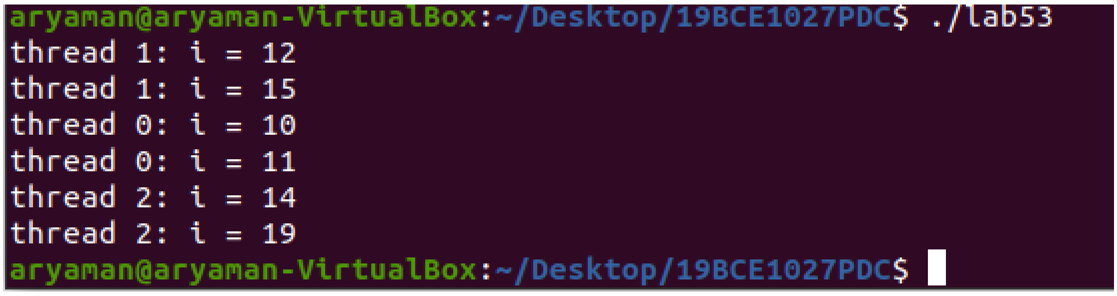
Exercise 2 Sample Problem on firstprivate

Code-static scheduling

```
#include <stdio.h>
#include <omp.h>
int main (void)
{
   int i = 10;
            int j;
            omp_set_num_threads(3);
   #pragma omp parallel for firstprivate(i) schedule(static,2)


                        for(j=0;j<6;j++){
      i = j+i;
```

```
                    printf("thread %d: i = %d\n", omp_get_thread_num(), i);


                    }

}
```

Output



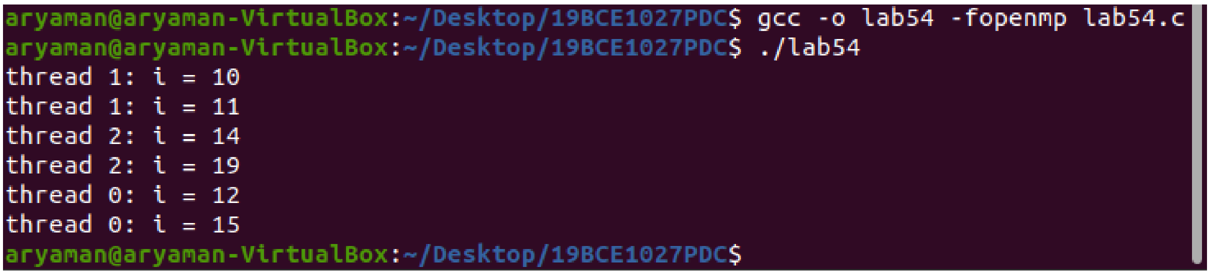
```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab53
thread 1: i = 12
thread 1: i = 15
thread 0: i = 10
thread 0: i = 11
thread 2: i = 14
thread 2: i = 19
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

Dynamic

CODE

```
#include <stdio.h>
#include <omp.h>
int main (void)
{
  int i = 10;
       int j;
       omp_set_num_threads(3);
  #pragma omp parallel for firstprivate(i) schedule(dynamic,2)


                    for(j=0;j<6;j++){
    i = j+i;
                    printf("thread %d: i = %d\n", omp_get_thread_num(), i);


                    }

}
```

OUTPUT



```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ gcc -o lab54 -fopenmp lab54.c
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab54
thread 1: i = 10
thread 1: i = 11
thread 2: i = 14
thread 2: i = 19
thread 0: i = 12
thread 0: i = 15
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

Exercise 3 (using Parallel For)

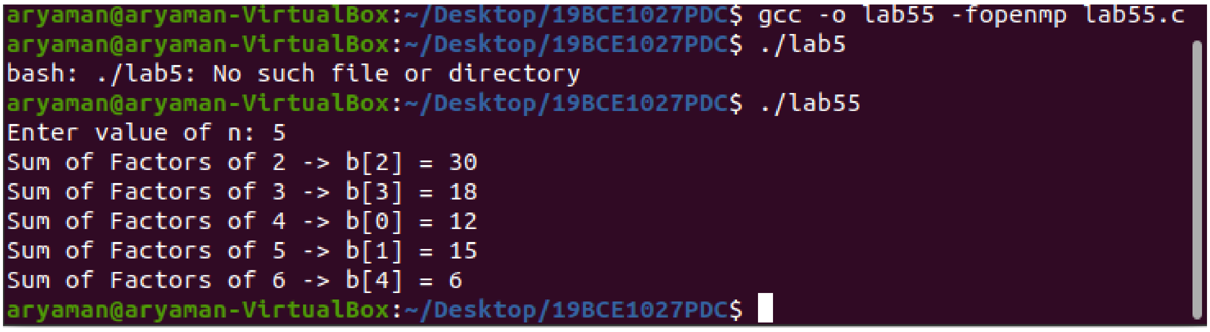
Addition of factors of a number < n

Static

Code

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

int main(void){
	int a[]={1,2,3,4,5,6,7,8,9,10};
	int n;
	printf("Enter value of n: ");
	scanf("%d",&n);
	int b[n];
	for(int i = 0; i < n; i++){
		b[i] = 0;
	}
	int j,k;
	int c = 2;
	omp_set_num_threads(6);
	#pragma omp parallel for schedule(static,2)
	for(k = 0; k < n; k++){

		for(j = 0; j < 10; j++)
		{
			if(a[j] % c == 0){
				b[k] += a[j];
			}
		}
		printf("Sum of Factors of %d -> b[%d] = %d\n",c,k,b[k]);

		c++;

	}
	return 0;
}
```

output



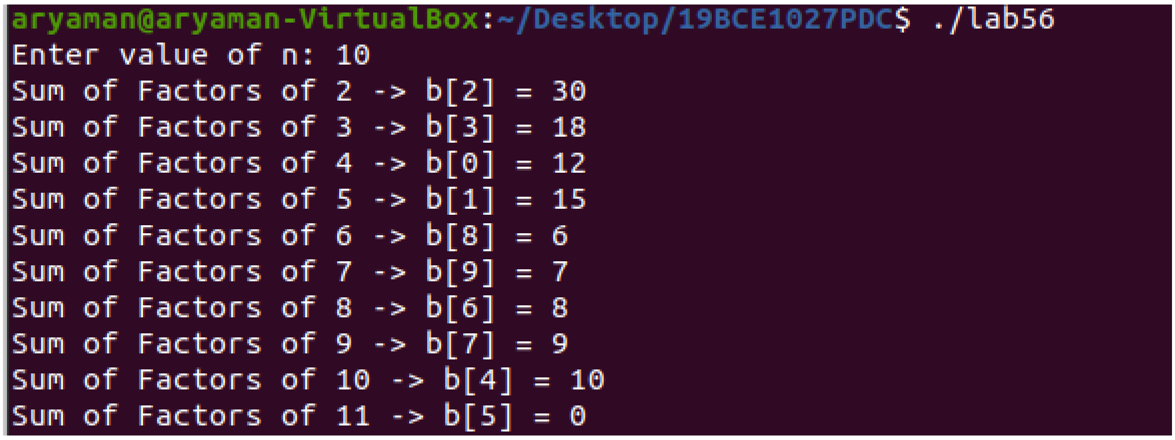
```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ gcc -o lab55 -fopenmp lab55.c
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab5
bash: ./lab5: No such file or directory
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab55
Enter value of n: 5
Sum of Factors of 2 -> b[2] = 30
Sum of Factors of 3 -> b[3] = 18
Sum of Factors of 4 -> b[0] = 12
Sum of Factors of 5 -> b[1] = 15
Sum of Factors of 6 -> b[4] = 6
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

Dynamic

Code

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

int main(void){
        int a[]={1,2,3,4,5,6,7,8,9,10};
        int n;
        printf("Enter value of n: ");
        scanf("%d",&n);
        int b[n];
        for(int i = 0; i < n; i++){
                b[i] = 0;
        }
        int j,k;
        int c = 2;
        omp_set_num_threads(6);
        #pragma omp parallel for schedule(dynamic,2)
        for(k = 0; k < n; k++){

                for(j = 0; j < 10; j++)
                {
                        if(a[j] % c == 0){
                                b[k] += a[j];
                        }
                }
                printf("Sum of Factors of %d -> b[%d] = %d\n",c,k,b[k]);

                c++;

        }
        return 0;
}
```

OUTPUT

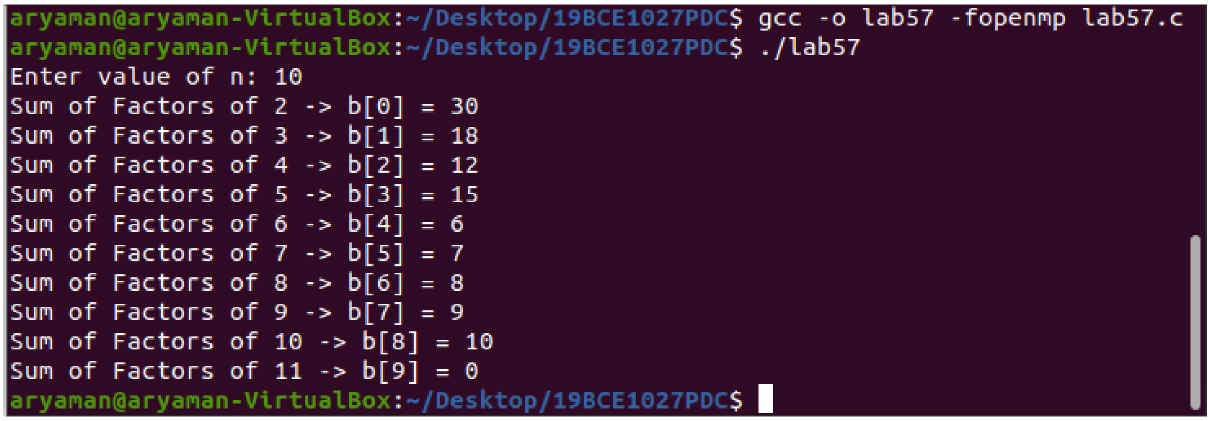


```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab56
Enter value of n: 10
Sum of Factors of 2 -> b[2] = 30
Sum of Factors of 3 -> b[3] = 18
Sum of Factors of 4 -> b[0] = 12
Sum of Factors of 5 -> b[1] = 15
Sum of Factors of 6 -> b[8] = 6
Sum of Factors of 7 -> b[9] = 7
Sum of Factors of 8 -> b[6] = 8
Sum of Factors of 9 -> b[7] = 9
Sum of Factors of 10 -> b[4] = 10
Sum of Factors of 11 -> b[5] = 0
```

<u>Guided</u>

Code

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

int main(void){
        int a[]={1,2,3,4,5,6,7,8,9,10};
        int n;
        printf("Enter value of n: ");
        scanf("%d",&n);
        int b[n];
        for(int i = 0; i < n; i++){
                b[i] = 0;
        }
        int j,k;
        int c = 2;
        omp_set_num_threads(6);
        #pragma omp parallel for schedule(guided,2)
        for(k = 0; k < n; k++){

                for(j = 0; j < 10; j++)
                {
                        if(a[j] % c == 0){
                                b[k] += a[j];
                        }
                }
                printf("Sum of Factors of %d -> b[%d] = %d\n",c,k,b[k]);

                c++;

        }
        return 0;
}
```



```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ gcc -o lab57 -fopenmp lab57.c
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab57
Enter value of n: 10
Sum of Factors of 2 -> b[0] = 30
Sum of Factors of 3 -> b[1] = 18
Sum of Factors of 4 -> b[2] = 12
Sum of Factors of 5 -> b[3] = 15
Sum of Factors of 6 -> b[4] = 6
Sum of Factors of 7 -> b[5] = 7
Sum of Factors of 8 -> b[6] = 8
Sum of Factors of 9 -> b[7] = 9
Sum of Factors of 10 -> b[8] = 10
Sum of Factors of 11 -> b[9] = 0
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ 
```

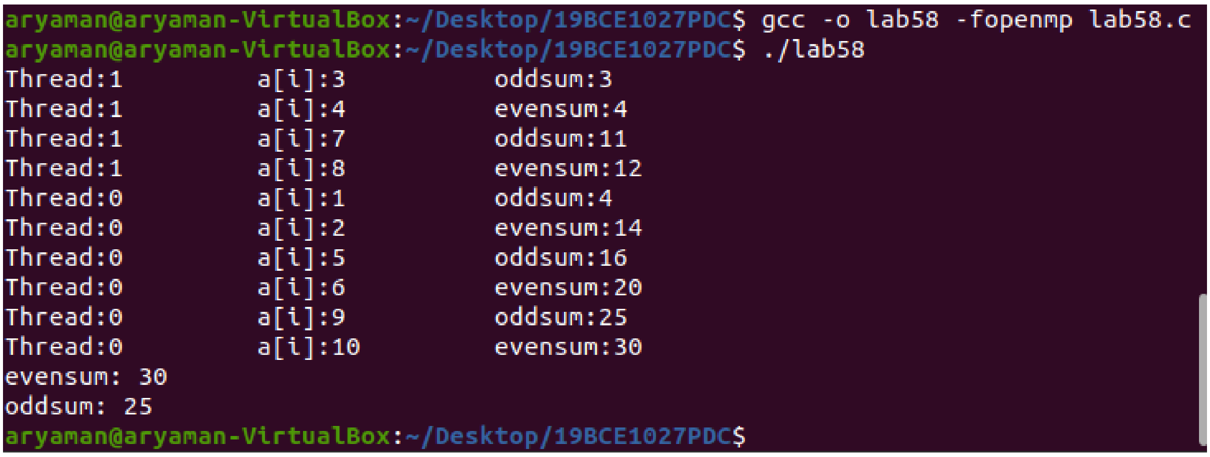
Exercise 4

addition of odd and even nos

code

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

void main()
{

        int a[]={1,2,3,4,5,6,7,8,9,10};
        int i,k;
        int oddsum=0;
        int evensum=0;

        omp_set_num_threads(2);
        #pragma omp parallel for schedule(static,2)

                for(i=0;i<10;i++){
                        if(a[i]%2==0){
                                evensum+=a[i];
```

```
                                        printf("Thread:%d\t a[i]:%d \t
evensum:%d\n",omp_get_thread_num(),a[i],evensum);
                              }
                              else{
                                        oddsum+=a[i];

                              printf("Thread:%d\t a[i]:%d \t
oddsum:%d\n",omp_get_thread_num(),a[i],oddsum);
                              }
                    }

          printf("evensum: %d\n",evensum);
          printf("oddsum: %d\n",oddsum);
}
```



```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ gcc -o lab58 -fopenmp lab58.c
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab58
Thread:1         a[i]:3          oddsum:3
Thread:1         a[i]:4          evensum:4
Thread:1         a[i]:7          oddsum:11
Thread:1         a[i]:8          evensum:12
Thread:0         a[i]:1          oddsum:4
Thread:0         a[i]:2          evensum:14
Thread:0         a[i]:5          oddsum:16
Thread:0         a[i]:6          evensum:20
Thread:0         a[i]:9          oddsum:25
Thread:0         a[i]:10         evensum:30
evensum: 30
oddsum: 25
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$
```

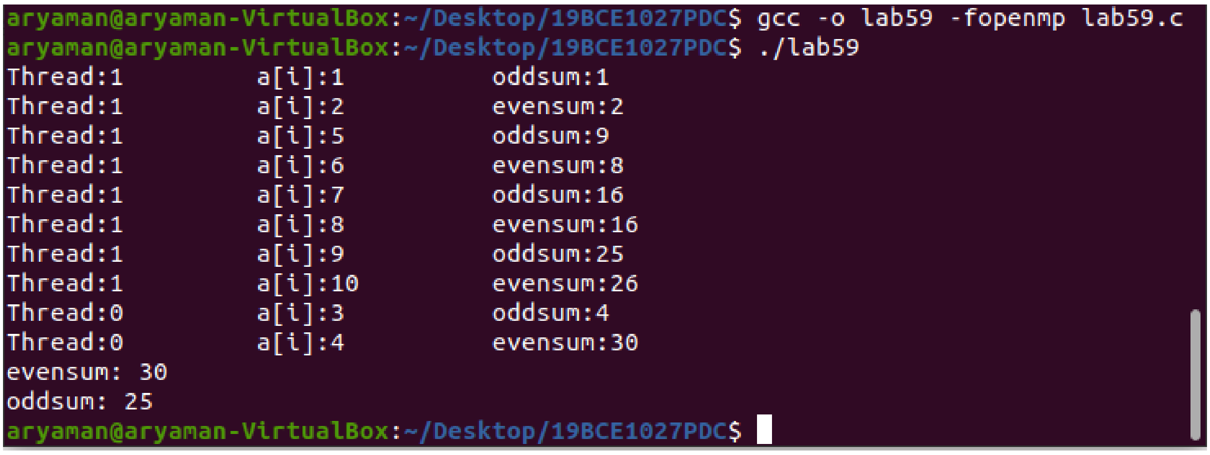
Dynamic

Code

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

void main()
{

          int a[]={1,2,3,4,5,6,7,8,9,10};
          int i,k;
          int oddsum=0;
          int evensum=0;

          omp_set_num_threads(2);
          #pragma omp parallel for schedule(dynamic,2)
```

```
        for(i=0;i<10;i++){
            if(a[i]%2==0){
                evensum+=a[i];

                printf("Thread:%d\t a[i]:%d \t
evensum:%d\n",omp_get_thread_num(),a[i],evensum);
            }
            else{
                oddsum+=a[i];

            printf("Thread:%d\t a[i]:%d \t
oddsum:%d\n",omp_get_thread_num(),a[i],oddsum);
            }
        }

    printf("evensum: %d\n",evensum);
    printf("oddsum: %d\n",oddsum);
}
```

OUTPUT



```
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ gcc -o lab59 -fopenmp lab59.c
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ ./lab59
Thread:1         a[i]:1          oddsum:1
Thread:1         a[i]:2          evensum:2
Thread:1         a[i]:5          oddsum:9
Thread:1         a[i]:6          evensum:8
Thread:1         a[i]:7          oddsum:16
Thread:1         a[i]:8          evensum:16
Thread:1         a[i]:9          oddsum:25
Thread:1         a[i]:10         evensum:26
Thread:0         a[i]:3          oddsum:4
Thread:0         a[i]:4          evensum:30
evensum: 30
oddsum: 25
aryaman@aryaman-VirtualBox:~/Desktop/19BCE1027PDC$ 
```