

Parallel and Distribution Lab Experiment 2

Name:Om Ashish Mishra

Registration Number: 16BCE0789

Slot: B2

The Question:

1. Write an OpenMP program to find the vector addition and dot product and parallelize using simple for loop
2. Write a C program and parallelize it using OpenMP Sections construct for the following scenario
a. Biggest of n numbers b. Smallest of n numbers c. Factorial of n d. Fibonacci sequence. and compute its execution time. Compare the execution time for sequential and parallel for different number of elements and tabulate the results for five entries.

The Answers:

1.

The Code:

```
void main()
{
    int a,i,b,x[100],y[100],s[100],m=0;

    printf("Enter the number of elements in the first vector : ");

    scanf("%d",&a);

    printf("Enter the elements in the first vector : \n");

    for(i=0;i<a;i++)
```

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

```
printf("Enter the number of elements in the second vector : ");
```

```
scanf("%d",&b);
```

```
printf("Enter the elements in the second vector : \n");
```

```
for(i=0;i<b;i++)
```

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

```
if (a == b)
```

```
{
```

```
    clock_t start_clock = clock();
```

```
#pragma omp parallel
```

```
{
```

```
    #pragma omp for
```

```
    for (i=0;i<a;i++)
```

```
    {
```

```
        s[i]=x[i]+y[i];
```

```
        m=m+x[i]*y[i];
```

```
    }
```

```
}
```

```
clock_t end_clock = clock();
```

```
printf("The Vector Sum is : ");
```

```
for(i=0;i<a;i++)
```

```
{
```

```

        printf("\n%d",s[i]);

    }

    printf("\nThe Vecor Dot Product is : %d",m);

    printf("\nThe Execution Time is : %ld ms",(end_clock-start_clock));

}

else

{

    printf("The Addition Not done!!");

}

}

```

The Output:

```

C:\Users\16BCE0789\Desktop\vector1.exe
Enter the number of elements in the first vector : 4
Enter the elements in the first vector :
1
2
3
4
Enter the number of elements in the second vector : 4
Enter the elements in the second vector :
1
2
3
4
The Vector Sum is :
2
4
6
8
The Vecor Dot Product is : 30
The Execution Time is : 1 ms
-----
Process exited after 15.8 seconds with return value 29
Press any key to continue . . .

```

2.

DONE WITH PARALLELISM

The Code:

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

```
#include<omp.h>

#include<time.h>

#include<math.h>

void main()

{

    int n,i,j, x[100],f=1,ch,max=0,min=1000,s=0;

    clock_t start, end;

    double k;

    printf("1. for Biggest in n numbers");

    printf("\n2. for Smallest in n numbers");

    printf("\n3. for Factorial of n");

    printf("\n4. for Fibonacci sequence");

    printf("\nEnter your choice : ");

    scanf("%d",&ch);

    printf("Enter the number of elements : ");

    scanf("%d",&n);

    if(ch == 1)

    {

        printf("Enter the elements into the array : \n");

        for(i=0;i<n;i++)

        {

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

        }

        start = clock();
```

```

#pragma omp parallel
{
    #pragma omp for
    for(i=0;i<n;i++)
    {
        if(max<x[i])
            max=x[i];
        else
            continue;
    }
}

end = clock();

k = ((double)(end-start))/CLOCKS_PER_SEC;

printf("The Largest Number is : %d",max);

printf("\nThe Execution Time is : %ld ms",k);
}

else if(ch == 2)
{
    printf("Enter the elements into the array : \n");

    for(i=0;i<n;i++)
    {
        scanf("%d",&x[i]);
    }

    start = clock();

    #pragma omp parallel

```

```

    {

        #pragma omp for
        for(i=0;i<n;i++)
        {

            if(min>x[i])

                min=x[i];

            else

                continue;

        }

    }

    end = clock();

    printf("The Smallest Number is : %d",min);

    k = ((double)(end-start))/CLOCKS_PER_SEC;

    printf("\nThe Execution Time is : %f ms",k);

}

else if(ch == 3)

{

    start = clock();

    #pragma omp parallel

    {

        #pragma omp for

        for(i=1;i<=n;i++)

        {

            f=1;

            for(j=1;j<=i;j++)

```

```

        {
            f=f*j;
        }
        s+=f;
    }
}

end = clock();

printf("The Factorial of %d numbers is : %d",n,s);

k = ((double)(end-start))/CLOCKS_PER_SEC;

printf("\nThe Execution Time is : %f ms",k);
}

else if(ch == 4)
{
    start = clock();

    int f1 = 0, f2 = 1;

    printf("The Fibonacci Series for n numbers is: %d, %d",f1,f2);

    #pragma omp parallel
    {
        #pragma omp for
        for(i=1;i<=n-2;i++)
        {
            s = f1+f2;

            f1=f2;

            f2=s;

            printf(", %d",s);

```

```

        }

    }

    end = clock();

    k = ((double)(end-start))/CLOCKS_PER_SEC;

    printf("\nThe Execution Time is : %f ms",k);

}

else

{

    printf("\nWrong Choice");

}

}

```

The Output:

1.

```

C:\Users\OM\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 1
Enter the number of elements : 7
Enter the elements into the array :
34
78
23
45
6
89
12
The Largest Number is : 89
The Execution Time is : 0.001000 ms
-----
Process exited after 30.13 seconds with return value 36
Press any key to continue . . .

```

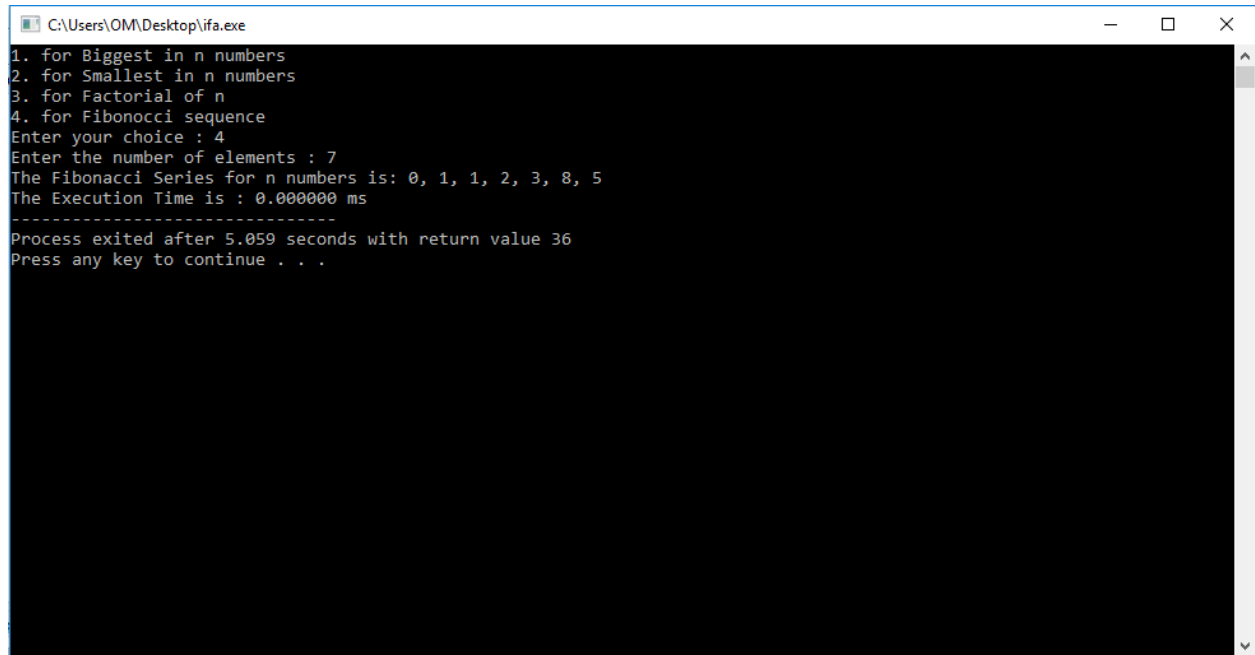
2.


```
C:\Users\OM\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 2
Enter the number of elements : 7
Enter the elements into the array :
34
45
56
7
23
78
12
The Smallest Number is : 7
The Execution Time is : 0.001000 ms
-----
Process exited after 18.56 seconds with return value 36
Press any key to continue . . .
```

3.

```
C:\Users\OM\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 3
Enter the number of elements : 19
The Factorial of 19 numbers is : 109641728
The Execution Time is : 0.000000 ms
-----
Process exited after 7.861 seconds with return value 36
Press any key to continue . . .
```

4.



```
C:\Users\OM\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 4
Enter the number of elements : 7
The Fibonacci Series for n numbers is: 0, 1, 1, 2, 3, 8, 5
The Execution Time is : 0.000000 ms
-----
Process exited after 5.059 seconds with return value 36
Press any key to continue . . .
```

DONE WITH NON PARALLELISM

The Code:

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

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

```
#include<time.h>
```

```
void main()
```

```
{
```

```
    int n,i,j, x[100],f=1,ch,max=0,min=1000,s=0;
```

```
    printf("1. for Biggest in n numbers");
```

```
    printf("\n2. for Smallest in n numbers");
```

```
    printf("\n3. for Factorial of n");
```

```
    printf("\n4. for Fibonacci sequence");
```

```
printf("\nEnter your choice : ");

scanf("%d",&ch);

printf("Enter the number of elements : ");

scanf("%d",&n);

if(ch == 1)
{
    printf("Enter the elements into the array : \n");

    for(i=0;i<n;i++)
    {
        scanf("%d",&x[i]);
    }

    clock_t start_clock = clock();

    #pragma omp for
    for(i=0;i<n;i++)
    {
        if(max<x[i])
            max=x[i];
        else
            continue;
    }

    clock_t end_clock = clock();

    printf("The Largest Number is : %d",max);

    printf("\nThe Execution Time is : %ld ms",(end_clock-start_clock));
```

```

    }
else if(ch == 2)
{
    printf("Enter the elements into the array : \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&x[i]);
    }
    clock_t start_clock = clock();

    #pragma omp for
    for(i=0;i<n;i++)
    {
        if(min>x[i])
            min=x[i];
        else
            continue;
    }

    clock_t end_clock = clock();
    printf("The Smallest Number is : %d",min);
    printf("\nThe Execution Time is : %ld ms",(end_clock-start_clock));
}
else if(ch == 3)
{

```

```
clock_t start_clock = clock();
```

```
#pragma omp for
```

```
for(i=1;i<=n;i++)
```

```
{
```

```
    f=1;
```

```
    for(j=1;j<=i;j++)
```

```
    {
```

```
        f=f*j;
```

```
    }
```

```
    s+=f;
```

```
}
```

```
clock_t end_clock = clock();
```

```
printf("The Factorial of %d numbers is : %d",n,s);
```

```
printf("\nThe Execution Time is : %ld ms", (end_clock-start_clock));
```

```
}
```

```
else if(ch == 4)
```

```
{
```

```
    clock_t start_clock = clock();
```

```
    int f1 = 0, f2 = 1;
```

```
    printf("The Fibonacci Series for n numbers is: %d, %d",f1,f2);
```

```
#pragma omp for
```

```
for(i=1;i<=n-2;i++)
```

```

        {

            s = f1+f2;

            f1=f2;

            f2=s;

            printf(", %d",s);

        }

        clock_t end_clock = clock();

        printf("\nThe Execution Time is : %ld ms", (end_clock-start_clock));

    }

    else

    {

        printf("\nWrong Choice");

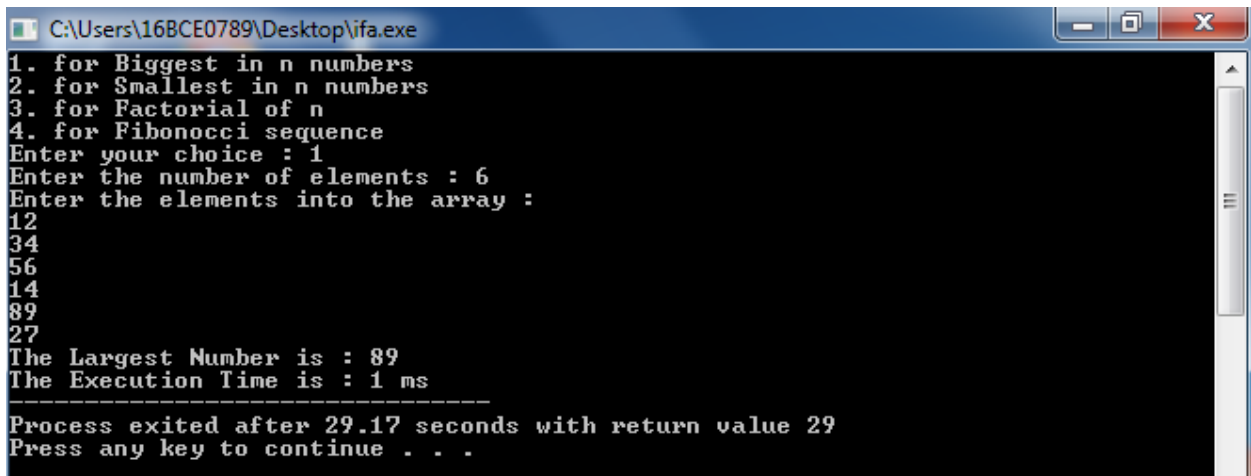
    }

}

```

The Output:

1. Largest Number

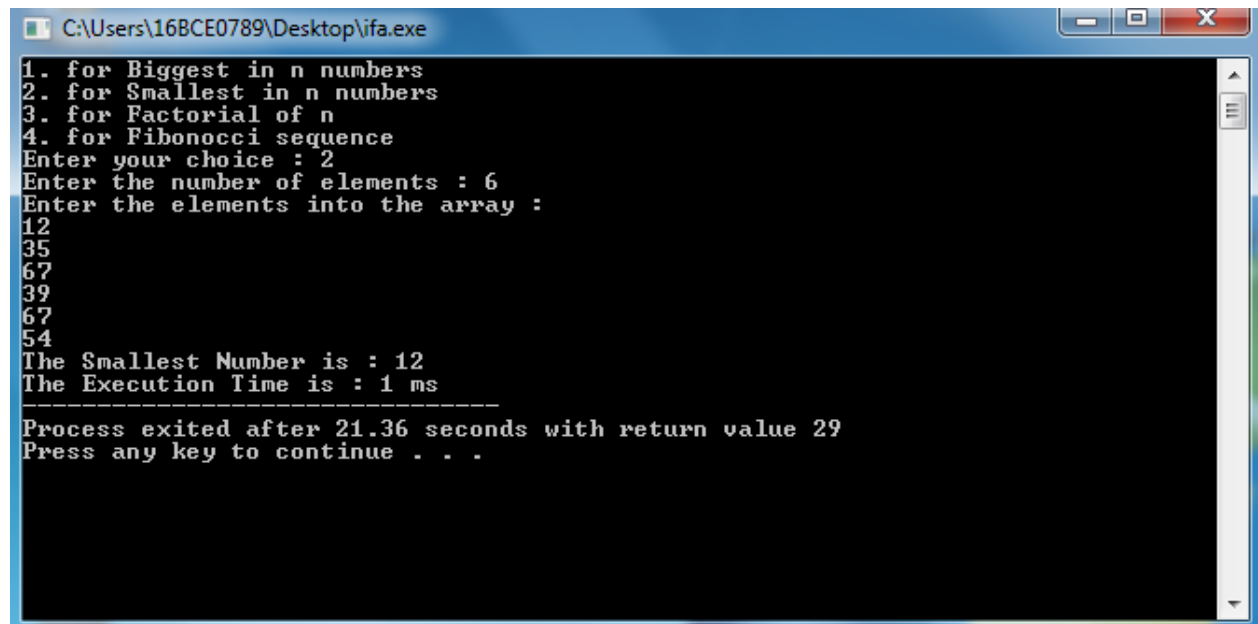


```

C:\Users\16BCE0789\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 1
Enter the number of elements : 6
Enter the elements into the array :
12
34
56
14
89
27
The Largest Number is : 89
The Execution Time is : 1 ms
-----
Process exited after 29.17 seconds with return value 29
Press any key to continue . . .

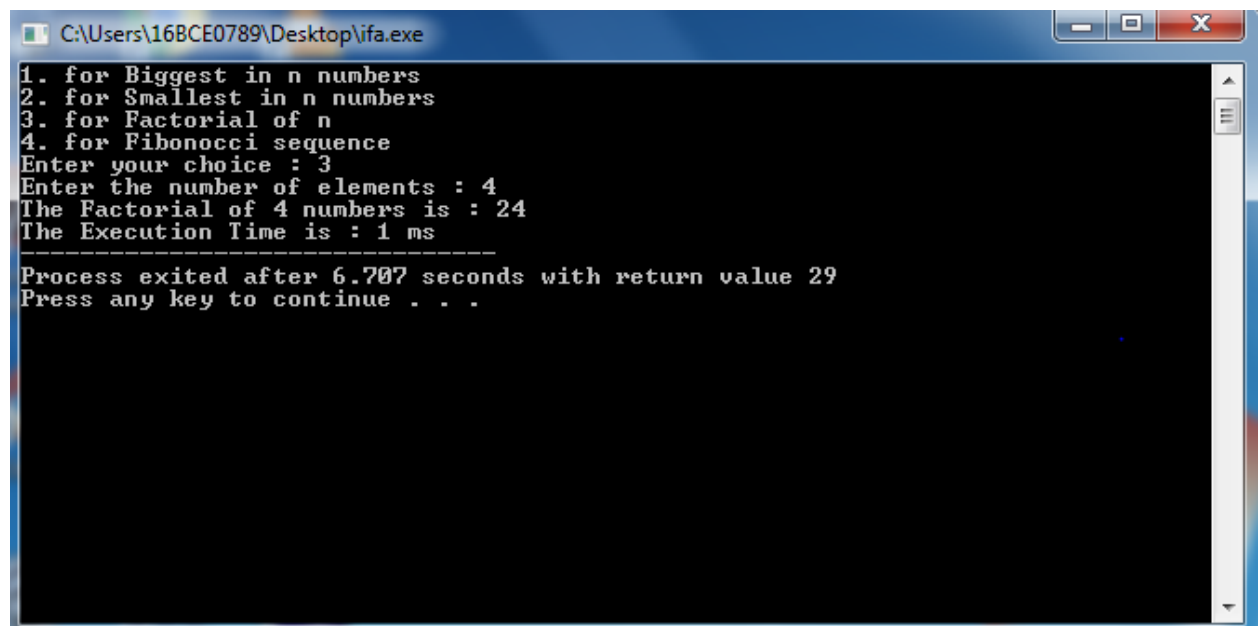
```

2. Smallest Number:



```
C:\Users\16BCE0789\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 2
Enter the number of elements : 6
Enter the elements into the array :
12
35
67
39
67
54
The Smallest Number is : 12
The Execution Time is : 1 ms
-----
Process exited after 21.36 seconds with return value 29
Press any key to continue . . .
```

3. Factorial Number:



```
C:\Users\16BCE0789\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 3
Enter the number of elements : 4
The Factorial of 4 numbers is : 24
The Execution Time is : 1 ms
-----
Process exited after 6.707 seconds with return value 29
Press any key to continue . . .
```

4. Fibonacci Number Sum:

```
C:\Users\OM\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 4
Enter the number of elements : 9
The Fibonacci Series for n numbers is: 0, 1, 1, 2, 13, 8, 3, 21, 5
The Execution Time is : 16 ms
-----
Process exited after 7.236 seconds with return value 30
Press any key to continue . . .
```

5. Wrong Choice:

```
C:\Users\16BCE0789\Desktop\ifa.exe
1. for Biggest in n numbers
2. for Smallest in n numbers
3. for Factorial of n
4. for Fibonacci sequence
Enter your choice : 5
Enter the number of elements : 4

Wrong Choice
-----
Process exited after 5.458 seconds with return value 13
Press any key to continue . . .
```

The Time Comparison:

Choice	Parallel	Serial
Largest Number	0.001000ms	1ms
Smallest Number	0.001000ms	1ms
Factorial	0ms	1ms
Fibonacci	0ms	16ms