



# **Lab Submission – 04**

Arnab Mondal

20BCE1294

**Program: B.Tech**

**Semester: Fall 2022-23**

**Course: CSE4001 – Parallel and Distributed Computing**

**Faculty: Dr. Sudha A**

**Date: 11-09-2022**

**Exercise: 04**

1. Perform calculator with four basic arithmetic operations that covers add, sub, multiply and modulo on each operation that is carried by different threads using sections.

Code:

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

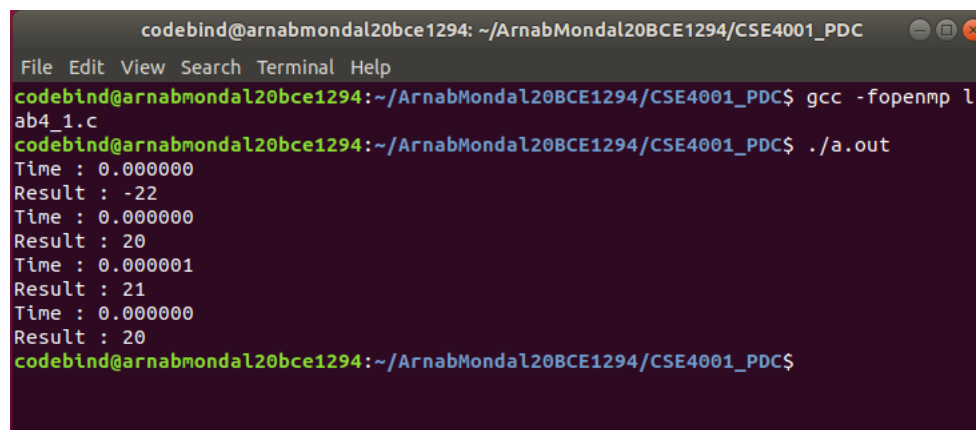
int main()
{
    int n1= 2147483647, n2 = 2147483627;
    double st, en;
    #pragma omp parallel
    {
        #pragma omp sections
        {
            #pragma omp section
            {
                int res1 = n1 + n2;
                printf("Time : %f\n", (en - st));
                printf("Result : %d\n", res1);
            }
        }
    }
}
```

```

    }
    #pragma omp section
    {
        int res2 = n1 - n2;
        printf("Time : %f\n", (en - st));
        printf("Result : %d\n", res2);
    }
    #pragma omp section
    {
        int res3 = n1 * n2;
        printf("Time : %f\n", (en - st));
        printf("Result : %d\n", res3);
    }
    #pragma omp section
    {
        int res4 = n1 % n2;
        printf("Time : %f\n", (en - st));
        printf("Result : %d\n", res4);
    }
}
return 0;
}

```

Output:



```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294/CSE4001_PDC
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ gcc -fopenmp l
ab4_1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ ./a.out
Time : 0.000000
Result : -22
Time : 0.000000
Result : 20
Time : 0.000001
Result : 21
Time : 0.000000
Result : 20
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$

```

2. Find the class students' subject wise total marks in any 6 six subjects. utilize array and reduction ensure the variable clauses (for ex: private /shared).

Code:

```

#include<stdio.h>
#include<omp.h>
int main()
{

```

```

int arr[6][6];
int i,j;
int sum = 0;
for(i = 0;i < 6;i++)
{
    sum += 10;
    for(j = 0;j < 6;j++)
    {
        //printf("Enter %d student's marks:",i + 1);
        //scanf("%d",&arr[i][j]);
        arr[i][j] = sum;
        sum += 2;
    }
}
int mark1, mark2, mark3, mark4, mark5, mark6;
mark1=mark2=mark3=mark4=mark5=mark6=0;
sum = 0;
#pragma omp parallel for reduction(+ : mark1) reduction(+ : mark2)
reduction(+ : mark3) reduction(+ : mark4) reduction(+ : mark5) reduction(+ :
mark6) shared(arr) private(i) ordered
    for(i = 0;i < 6;i++)
    {
        mark1 += arr[i][0];
        mark2 += arr[i][1];
        mark3 += arr[i][2];
        mark4 += arr[i][3];
        mark5 += arr[i][4];
        mark6 += arr[i][5];
    }
    printf("Total of mark1 : %d\n",mark1);
    printf("Total of mark2 : %d\n",mark2);
    printf("Total of mark3 : %d\n",mark3);
    printf("Total of mark4 : %d\n",mark4);
    printf("Total of mark5 : %d\n",mark5);
    printf("Total of mark6 : %d\n",mark6);
return 0;
}

```

Output:

```
codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294/CSE4001_PDC
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ gcc -fopenmp l
ab4_2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ ./a.out
Total of mark1 : 390
Total of mark2 : 402
Total of mark3 : 414
Total of mark4 : 426
Total of mark5 : 438
Total of mark6 : 450
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$
```

3. Write the producer consumer problem using critical section directives.

Code:

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

int mutex = 1;
int full = 0;

int empty = 10, x = 0;

void producer()
{
    --mutex;
    ++full;
    --empty;
    x++;
    printf("\nProducer produces item %d", x);
    ++mutex;
}

void consumer()
{
    --mutex;
    --full;
    ++empty;
    printf("\nConsumer consumes item %d", x);
    x--;
    ++mutex;
}

int main()
{
    int n, i;
```

```

printf("\n1.Producer"
      "\n2.Consumer"
      "\n3.Exit");
#pragma omp critical

for (i = 1; i > 0; i++) {

    //printf("\n1.Producer"
    //"\n2.Consumer"
    //"\n3.Exit");
    printf("\nEnter any of the above options : ");
    scanf("%d", &n);

    switch (n) {
    case 1:
        if ((mutex == 1)
            && (empty != 0)) {
            producer();
        }

        else {
            printf("Buffer is full");
        }
        break;

    case 2:

        if ((mutex == 1)
            && (full != 0)) {
            consumer();
        }

        else {
            printf("Buffer is empty");
        }
        break;

    case 3:
        exit(0);
        break;
    }
}
}

```

Output:

```
codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294/CSE4001_PDC
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ gedit lab4_3.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ gcc -fopenmp lab4_3.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$ ./a.out

1.Producer
2.Consumer
3.Exit
Enter any of the above options : 1

Producer produces item 1
Enter any of the above options : 1

Producer produces item 2
Enter any of the above options : 2

Consumer consumes item 2
Enter any of the above options : 2

Consumer consumes item 1
Enter any of the above options : 2
Buffer is empty
Enter any of the above options : 3
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294/CSE4001_PDC$
```