

**Name** : **Ibadullah Shaikh**

**Roll No** : **19k-0259**

**Section** : **BCS-5G**

### Lab 08

---

#### Q1.

#### Code:

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

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

```
int A[3][3]={1,5,10},{15,20,25},{30,35,40}};
```

```
int B[3][3]={1,2,3},{4,5,6},{7,8,9}};
```

```
int C[3][3]={0};
```

```
int main()
```

```
{
```

```
    #pragma omp parallel
```

```
    #pragma omp for
```

```
    for(int i=0;i<3*3;i++)
```

```
    {
```

```
        *(&C[i])=*(&A[i])+*(&B[i]);
```

```
        printf("%d\n",i);
```

```
    }
```

```
    printf("threads: %d\n",omp_get_num_threads());
```

```
    for(int i=0;i<3*3;i++)
```

```
    {
```

```
        if(i%3==0){
```

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

```

        }
        printf(" %d ",*(C+i));
    }
return 0;
}

```

---

## Q2.

### Code.

```

#include<stdio.h>

#include<omp.h>

int A[3][3]={1,5,10},{15,20,25},{30,35,40}};
int B[3][3]={1,2,3},{4,5,6},{7,8,9}};
int C[3][3]={0};

int multiply(int i,int j)
{
    int sum=0;
    for(int k=0;k<3;k++)
    {
        sum=sum+(A[i][k]*B[k][j]);
    }
return sum;
}

int main()
{
    #pragma omp parallel
    #pragma omp for

```

```

    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            C[i][j] = multiply(i , j);
        }
    }
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            printf(" %d ",C[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

---

### Q3.

#### Code.

```

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

long double x, factorial[100], power[100], result[100], sum=0;
int i, term;

long double FACTORIAL(long double temp)
{

```

```

        if(temp<=1)
        {
            return 1;
        }
        return temp * FACT(temp-1);
    }
}

long double POWER(double num, int pwr)
{
    long double res=1;
    for(int i=0;i<pwr;i++)
    {
        res = res * num;
    }
    return res;
}

int main()
{
    double x = 5;

    #pragma omp parallel
    #pragma omp for
    for(int i = 0 ; i < 15 ; i++)
    {
        factorial[i] = FACTORIAL(i);
        power[i] = i;
        result[i] = POWER(x,(int)power[i])/factorial[i] ;
        printf("Term %d %Lf\n",i+1,result[i]);
        sum=sum+result[i];
    }

    printf("Total Value of e^%f = %Lf + ",x,sum);
}

```

```
return 0;
```

```
}
```