

Name: Muhammad Usman

Roll no: 19P-0116

Assignment # 3

Code:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<pthread.h>
5
6 #define M 3
7 #define K 2
8 #define N 3
9
10
11 struct Matrix {
12     int A[M][K];
13     int B[K][N];
14     int C[M][N];
15 };
16
17
18 void *multiplyMatrix(void *ptr){
19     struct Matrix *myPtr = (struct Matrix*)ptr;
20
21     int sum = 0;
22     for(int i=0; i<M; i++)//multiplication loop
23     {
24         for(int j=0; j<N; j++)
25         {
26             for(int k=0; k<K; k++)
27             {
28                 sum=sum+ (*myPtr).A[i][k] * (*myPtr).B[k][j];
29             }
30             (*myPtr).C [i][j]=sum;
31             sum=0;
32         }
33     }
34     printf("Multiplication of A and B:\n");//printing the result Matrix (C)
35     for (int i = 0; i<M ; i++)
36     {
37         for (int j = 0; j < N; j++)
38         {
39             printf("%d \t",(*myPtr).C[i][j]);
40         }
41         printf("\n");
42     }
43 }
```

```

48
49 void assignValues()//assigns values to the structure
50 {
51     struct Matrix m;
52     int A[M][K] = {{1, 4}, {2, 5}, {3, 6}};
53     int B[K][N] = {{8, 7, 6}, {5, 4, 3}};
54     //Assigning values to Matrix A
55     for (int i = 0; i<M ; i++)
56     {
57         for (int j = 0; j < K; j++)
58         {
59             m.A[i][j] = A[i][j];
60         }
61     }
62     //Assigning values to Matrix B
63     for (int i = 0; i< K ; i++)
64     {
65         for (int j = 0; j < N; j++)
66         {
67             m.B[i][j] = B[i][j];
68         }
69     }
70
71     //Matrix C (this will be the result matrix)
72     for (int i = 0; i<M ; i++)
73     {
74         for (int j = 0; j < N; j++)
75         {
76             m.C[i][j] = 0;
77         }
78     }
79
80 }

```

```

81
82 //creating a pointer for object of Matrix
83 struct Matrix *ptr = &m;
84
85 pthread_t id;
86
87 //creating a thread and calling Matrix Multiplication function
88 pthread_create(&id, NULL, multiplyMatrix, (void*)ptr);
89
90 //joining the main thread
91 pthread_join(id, NULL);
92
93 }
94
95 int main(){
96
97     struct Matrix m;
98
99     assignValues(m);
100
101
102     return 0;
103 }

```

Output:



```
usman@usman-Inspiron-3520: ~/os_lab/os_labWork
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gedit osAssign3.c
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gcc -o osAssign3.o osAssign3.c -lpthread
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ ./osAssign3.o
Multiplication of A and B:
28      23      18
41      34      27
54      45      36
usman@usman-Inspiron-3520:~/os_lab/os_labWork$
```

The image shows a terminal window with a dark background. The window title is "usman@usman-Inspiron-3520: ~/os_lab/os_labWork". The user has executed three commands: "gedit osAssign3.c", "gcc -o osAssign3.o osAssign3.c -lpthread", and "./osAssign3.o". The output of the program is displayed, showing the text "Multiplication of A and B:" followed by a 3x3 matrix of numbers: 28, 23, 18; 41, 34, 27; 54, 45, 36. The terminal prompt is currently at "usman@usman-Inspiron-3520:~/os_lab/os_labWork\$".

