

1. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
int t_id[2] = {1, 2};
void *t_func(int *id);
int sum = 0;
pthread_mutex_t mutex;
int main() {
    pthread_t t[2];
    pthread_mutex_init(&mutex, NULL);
    for (int i = 0; i < 2; i++) {
        pthread_create(&t[i], NULL, (void *)t_func, &t_id[i]);
    }
    for (int i = 0; i < 2; i++) {
        pthread_join(t[i], NULL);
    }
    pthread_mutex_destroy(&mutex);
    printf("Total count: %d\n", sum);
    return 0;
}
void *t_func(int *id) {
    printf("Entered in Thread %d ...\n", *id);
    for (int i = 0; i < 3; i++) {
        pthread_mutex_lock(&mutex);
        sum++;
        pthread_mutex_unlock(&mutex);
    }
}
```

### Output:

```
Entered in Thread 1...
Entered in Thread 2...
Total count: 6
```

2. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
int t_id[2] = {1, 2};
void *t_func(int *id);
int sum = 0;
sem_t s;
int main() {
    pthread_t t[2];
    sem_init(&s, 0, 1);
    for (int i = 0; i < 2; i++) {
        pthread_create(&t[i], NULL, (void *)t_func, &t_id[i]);
    }
    for (int i = 0; i < 2; i++) {
        pthread_join(t[i], NULL);
    }
    sem_destroy(&s);
    printf("Total count: %d\n", sum);
    return 0;
}
void *t_func(int *id) {
    printf("Entered in Thread %d ...\n", *id);
    for (int i = 0; i < 3; i++) {
        sem_wait(&s);
        sum++;
        sem_post(&s);
    }
}
```

### Output:

```
Entered in Thread 1...
Entered in Thread 2...
Total count: 6
```

3. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
int t_id[] = {1, 2};
void *t_func1(int *id);
void *t_func2(int *id);
int sum = 15;
pthread_mutex_t m;
sem_t s;
int main() {
    pthread_t t[2];
    sem_init(&s, 0, 0);
    pthread_mutex_init(&m, NULL);
    pthread_create(&t[0], NULL, (void *)t_func1, &t_id[0]);
    pthread_create(&t[1], NULL, (void *)t_func2, &t_id[1]);
    for(int i = 0; i < 2; i++) {
        pthread_join(t[i], NULL);
    }
    sem_destroy(&s);
    pthread_mutex_destroy(&m);
    printf("Total sum: %d\n", sum);
    return 0;
}
void *t_func1(int *id) {
    sem_wait(&s);
    pthread_mutex_lock(&m);
    for(int i = 0; i < 5; i++) {
        printf("Sum: %d\n", sum);
        sum -= 10;
    }
    pthread_mutex_unlock(&m);
    sem_post(&s);
}
void *t_func2(int *id) {
    pthread_mutex_lock(&m);
    for (int i = 0; i < 5; i++) {
        printf("Sum: %d\n", sum);
        sum *= 3;
    }
    pthread_mutex_unlock(&m);
    sem_post(&s);
}
```

## Output:

```
Sum: 15
Sum: 45
Sum: 135
Sum: 405
Sum: 1215
Sum: 3645
Sum: 3635
Sum: 3625
Sum: 3615
Sum: 3605
Total sum: 3595
```

4. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
int t_id[] = {1, 2};
void *t_func1(int *id);
void *t_func2(int *id);
int sum = 0;
sem_t s1, s2;
int main() {
    pthread_t t[2];
    sem_init(&s1, 0, 1);
    sem_init(&s2, 0, 0);
    pthread_create(&t[0], NULL, (void *)t_func1, &t_id[0]);
    pthread_create(&t[1], NULL, (void *)t_func2, &t_id[1]);
    for(int i = 0; i < 2; i++) {
        pthread_join(t[i], NULL);
    }
    sem_destroy(&s1);
    sem_destroy(&s2);
    printf("Total sum: %d\n", sum);
    return 0;
}
void *t_func1(int *id) {
    sem_wait(&s1);
    for(int i = 0; i < 10; i++){
        printf("Sum: %d\n", sum);
        sum += 10;
    }
    sem_post(&s1);
    sem_post(&s2);
}
void *t_func2(int *id) {
    sem_wait(&s2);
    for(int i = 0; i < 10; i++) {
        printf("Sum: %d\n", sum);
        sum -= 5;
    }
    sem_post(&s2);
}
```

## Output:

```
Sum: 0
Sum: 10
Sum: 20
Sum: 30
Sum: 40
Sum: 50
Sum: 60
Sum: 70
Sum: 80
Sum: 90
Sum: 100
Sum: 95
Sum: 90
Sum: 85
Sum: 80
Sum: 75
Sum: 70
Sum: 65
Sum: 60
Sum: 55
Total sum: 50
```

5. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
int t_id[] = {1, 2, 3};
void *t_func1(int *id);
void *t_func2(int *id);
void *t_func3(int *id);
int sum = 13;
sem_t s1, s2, s3;
int main() {
    pthread_t t[3];
    sem_init(&s1, 0, 0);
    sem_init(&s2, 0, 1);
    sem_init(&s3, 0, 0);
    pthread_create(&t[0], NULL, (void *)t_func1, &t_id[0]);
    pthread_create(&t[1], NULL, (void *)t_func2, &t_id[1]);
    pthread_create(&t[2], NULL, (void *)t_func3, &t_id[2]);
    for(int i = 0; i < 3; i++) {
        pthread_join(t[i], NULL);
    }
    sem_destroy(&s1);
    sem_destroy(&s2);
    sem_destroy(&s3);
    printf("Total sum: %d\n", sum);
    return 0;
}
void *t_func1(int *id) {
    sem_wait(&s1);
    for(int i = 0; i < 5; i++) {
        printf("Sum: %d\n", sum);
        sum *= 2;
    }
    sem_post(&s1);
}
void *t_func2(int *id) {
    sem_wait(&s2);
    for(int i = 0; i < 5; i++) {
        printf("Sum: %d\n", sum);
        sum += 7;
    }
    sem_post(&s2);
    sem_post(&s3);
}
void *t_func3(int *id) {
    sem_wait(&s3);
    for(int i = 0; i < 5; i++) {
        printf("Sum: %d\n", sum);
        sum -= 3;
    }
    sem_post(&s3);
    sem_post(&s1);
}
```

## Output:

```
Sum: 15
Sum: 45
Sum: 135
Sum: 405
Sum: 1215
Sum: 3645
Sum: 3635
Sum: 3625
Sum: 3615
Sum: 3605
Total sum: 3595
```



6. Find the outputs of the following code. [Run this code using any IDE multiple times and analyse the outputs]

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
int t_id[] = {1, 2, 3, 4};
void *func(int *id);
int p = 45;
int q = 78;
sem_t s1, s2, s3, s4;
int main() {
    pthread_t t[4];
    sem_init(&s1, 0, 0);
    sem_init(&s2, 0, 0);
    sem_init(&s3, 0, 0);
    sem_init(&s4, 0, 1);
    for (int i = 0; i < 4; i++) {
        pthread_create(&t[i], NULL, (void *)func, &t_id[i]);
    }
    for (int i = 0; i < 4; i++) {
        pthread_join(t[i], NULL);
    }
    sem_destroy(&s1);
    sem_destroy(&s2);
    sem_destroy(&s3);
    sem_destroy(&s4);
    printf("Final p: %d\nFinal q: %d\n", p, q);
    return 0;
}

void *func(int *id) {
    if (*id == 1) {
        sem_wait(&s1);
        printf("p: %d\nq: %d\n", p, q);
        p = p - 15;
        q = q - p;
        sem_post(&s1);
    } else if (*id == 2) {
        sem_wait(&s3);
        printf("p: %d\nq: %d\n", p, q);
        p = p + q;
        q = p - 7;
        sem_post(&s3);
        sem_post(&s1);
    } else if (*id == 3) {
        sem_wait(&s4);
        printf("p: %d\nq: %d\n", p, q);
        p = p * 9;
        q = q * 11;
        sem_post(&s4);
        sem_post(&s2);
    } else {
        sem_wait(&s2);
        printf("p: %d\nq: %d\n", p, q);
        p = p + q;
        q = p + q;
        sem_post(&s2);
        sem_post(&s3);
    }
}
```

## Output:

```
p: 45  
q: 78  
p: 405  
q: 858  
p: 1263  
q: 2121  
p: 3384  
q: 3377  
Final p: 3369  
Final q: 8
```