

1. Find outputs of the following code. [Run this code in the PC 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=1;
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);
}

```

mutex lock { available = T
 acquire
 !T = F
 available = F
 unlock
 available = T

Queue
 Semaphore
 value = 0
 blocked

Sum: 15
 Sum: 45
 Sum: 135
 Sum: 405
 Sum: 1215
 Sum: 3645

Sum: 3635
 Sum: 3625
 Sum: 3615
 Sum: 3605

Total sum: 3595

0 ← thread
 1 ← Value
 Initialize

value

s = 0 - 1

signal

0

s = 0

wake up

1

2. Find outputs of the following code. [Run this code in the PC 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);
}
```

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

$s_1 = 2, 0, 1$
 $s_2 = 0, 1, 0$

$v=1$ (unblocked)
open

7

3. Find outputs of the following code. [Run this code in the PC 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; 98 38 1056
✓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);
}

```

Sum: 13
Sum: 20
Sum: 27
Sum: 34
Sum: 41

Sum: 48
Sum: 45
Sum: 42
Sum: 39
Sum: 36

Sum: 33
Sum: 66
Sum: 132
Sum: 264
Sum: 528

Total sum: 1056

