EXERCICES

1. Synchronize the following processes to display a message: “Em yêu anh không”

```
P1 {
    printf(“Em”);
}
```

```
P2 {
    printf(“Yêu”);
}
```

```
P3 {
    printf(“Anh”);
}
```

```
P4 {
    printf(“Không”);
}
```

EXERCICES

1. Synchronize the following processes to display a message: “Em yêu anh không”

```
semaphore s1 = 0, s2 = 0, s3 = 0
```

```
P1 {
    printf(“Em”);
    up(s1);
}
```

```
P2 {
    down(s1);
    printf(“Yêu”);
    up(s2);
}
```

```
P3 {
    down(s2);
    printf(“Anh”);
    up(s3);
}
```

```
P4 {
    down(s3);
    printf(“Không”);
}
```

Em yêu anh không. Em yêu anh không. Em yêu anh không ...

EXERCICES

2. Synchronize the following processes to display a message: “Anh không yêu em”

```
P1 {
    printf(“Em”);
}
```

```
P2 {
    printf(“Yêu”);
}
```

```
P3 {
    printf(“Anh”);
}
```

```
P4 {
    printf(“Không”);
}
```

EXERCICES

2. Synchronize the following processes to display messages: “Anh không yêu em”

```
semaphore        s2 = 0, s3 = 0, s4 = 0
```

```
P1 {
    down(s2);
    printf(“Em”);
    up(s1);
}
```

```
P2 {
    down(s4);
    printf(“Yêu”);
    up(s2);
}
```

```
P3 {
    down(s1);
    printf(“Anh”);
    up(s3);
}
```

```
P4 {
    down(s3);
    printf(“Không”);
    up(s4);
}
```

Anh không yêu em. Anh không yêu em. Anh không yêu em. ...

EXERCICES

3. Consider two processes A et B as follows.

```
Process A {
    while(TRUE) {

        na = na + 1;

}}
```

```
Process B {
    while(TRUE) {

        nb = nb + 1;

}}
```

Synchronize A and B to ensure:

a. na <= nb + 10 at any time

b. nb < na <= nb + 10 at any time

EXERCICES

3. Consider two processes A et B as follows.

```
semaphore sB = 10
```

```
Process A {
    while(TRUE) {
        down(sB);
        na = na + 1;

}}
```

```
Process B {
    while(TRUE) {

        nb = nb + 1;
        up(sB);
}}
```

Synchronize A and B to ensure:

a. na <= nb + 10 at any time

EXERCICES

3. Consider two processes A et B as follows.

```
semaphore sB = 10, sA =
```

```
Process A {
    while(TRUE) {
        down(sB);
        na = na + 1;
        up(sA);
}}
```

```
Process B {
    while(TRUE) {
        down(sA);
        nb = nb + 1;
        up(sB);
}}
```

Synchronize A and B to ensure:

b. nb < na <= nb + 10 at any time

EXERCICES

4. Consider three processes A, B, and C as follows.

Semaphore s1=…, s2=…, s3=…, s4=…, s5=…, s6=…

```
Process A {
    task1();
    task4();
}
```

```
Process B {
    task2();
    task5();
}
```

```
Process C {
    task3();
    task6();
}
```

Synchronize these three processes to ensure execution order as follows: task1 → task2 → ... → task6 → task1 → task2

EXERCICES

4. Consider three processes A, B, and C as follows.

Semaphore s1=0, s2=0, s3=0, s4=0, s5=0, s6=1

```
Process A {
    Down(s6)
    task1();
    Up(s1)
    Down(s3)
    task4();
    Up(s4)
}
```

```
Process B {
    Down(s1)
    task2();
    Up(s2)
    Down(s4)
    task5();
    Up(s5)
}
```

```
Process C {
    Down(s2)
    task3();
    Up(s3)
    Down(s5)
    task6();
    Up(s6)
}
```

Synchronize these three processes to ensure execution order as follows: task1 → task2 → ... → task6 → task1 → task2