

①

Name - Ankit Goyal

Roll No - 17103011

Subject - AOS

Subject code - CSX-352

Q=37

```
#include <iostream>
```

```
#include <unistd.h>
```

```
#include <pthread.h>
```

```
using namespace std;
```

```
class Monitor {
```

```
private: int rcnt, wcnt, waitr, waitw;
```

```
pthread_cond_t canread, canwrite, candlock;
```

```
public:
```

```
monitor()
```

```
{
```

```
rcnt rcnt = wcnt = waitr = waitw = 0;
```

```
pthread_cond_init(&canread, NULL);
```

```
pthread_cond_init(&canwrite, NULL);
```

```
pthread_cond_init(&candlock, NULL);
```

```
}
```

```
void beginread
```

```
void beginread (int i)
```

```
{
```

```
pthread_mutex_lock(&candlock);
```

```
if (wcnt == 1) {
```

```
waitr++;
```

```
pthread_cond_wait(&canread, &candlock);
```

```
waitr--;
```

```
}
```

```
rcnt++;
```

```
cout << "Reader " << i << " is reading";
```

```
pthread_mutex_unlock(&candlock);
```

```
pthread_cond_broadcast(&canread);
```

```
}
```

②

```
void endread (int i)
{
    pthread_mutex_lock (&condlock);
    if (--rct == 0)
    {
        pthread_cond_signal (&canwrite);
    }
    pthread_mutex_unlock (&condlock);
}

void beginwrite (int i)
{
    pthread_mutex_lock (&condlock);
    if (wcnt == 1 || rct > 0 || waitr > 0)
    {
        waitw++;
        pthread_cond_wait (&canwrite, &condlock);
        waitw--;
    }
    wcnt = 1;
    cout << "writer" << i << " is writing ";
    pthread_mutex_unlock (&condlock);
}

void endwrite (int i)
{
    pthread_mutex_lock (&condlock);
    wcnt = 0;
    if (waitr > 0)
        pthread_cond_signal (&canread);
    else
        pthread_cond_signal (&canwrite);
    pthread_mutex_unlock (&condlock);
}

} M;
```

```
void* reader (void* id)
{
    int c = 0;
    int i = *(int*) id;
    while (c < 5) {
        usleep(1);
        M.beginread (i);
        M.endread (i);
        c++;
    }
}
```

void * writer (void * id)

```
{
    int c=0;
    int i= (int) id;
    while (c<5)
    {
        usleep(1);
        m.beginwrite(i);
        m.endwrite(i);
        c++;
    }
}
```

int main()

```
{
    pthread_t r[5], w[5];
```

```
    int rd[5];
```

```
    for (int i=0; i<5; i++)
```

```
    {
        id[i]=i;
```

```
        pthread_create (&r[i], Null, &reader, &id[i]);
```

```
        pthread_create (&w[i], Null, &writer, &id[i]);
    }
```

```
}
```

```
for (int i=0; i<5; i++) {
```

```
    pthread_join(r[i], Null);
```

```
    pthread_join(w[i], Null);
}
```

```
}
```

```
}
```