Semaphore in process synchronisation

code

```c
// Nyasha Vasoya
// 202251155

#include <stdio.h>
#include <stdbool.h>

#define MAX_PROCESSES 10
#define QUEUE_SIZE 20


int processQueue[QUEUE_SIZE];
int qFront = 0;
int qRear = -1;
int qSize = 0;


void enqueue(int processId) {
    if (qSize >= QUEUE_SIZE) {
        printf("queue is full\n");
        return;
    }
    qRear = (qRear + 1) % QUEUE_SIZE;
    processQueue[qRear] = processId;
    qSize++;
}

int dequeue() {
    if (qSize <= 0) {
        return -1;
    }
    int processId = processQueue[qFront];
    qFront = (qFront + 1) % QUEUE_SIZE;
    qSize--;
    return processId;
}


void P(int* semaphore, int processId) {
    (*semaphore)--;
    if (*semaphore < 0) {
        enqueue(processId);
        printf("P%d is blocked\n", processId);
    } else {
        printf("P%d continues execution\n", processId);
    }
}

void V(int* semaphore) {
    (*semaphore)++;
    if (*semaphore <= 0) {
        int unblockedProcess = dequeue();
        if (unblockedProcess != -1) {
            printf("P%d is woken up\n", unblockedProcess);
        }
    } else {
        printf("No process to wake up\n");
    }
}


int main() {
    int semaphore = 1;

    P(&semaphore, 1);
    P(&semaphore, 2);
    V(&semaphore);
    P(&semaphore, 1);
    V(&semaphore);
    P(&semaphore, 3);
    V(&semaphore);

    return 0;
}
```

output:

```
Process 1 continues execution
Process 2 is blocked
Process 2 is woken up
Process 1 is blocked
Process 1 is woken up
Process 3 is blocked
Process 3 is woken up
```