

```

void departure(void) /* Departure event function. */
{
    int i;
    float delay;
    /* Check to see whether the queue is empty. */
    if (num_in_q == 0)
    {
        /* The queue is empty so make the server idle and eliminate the departure (service completion)
        event from consideration. */
        server_status = IDLE;
        time_next_event[2] = 1.0e+30;
    } else {
        /* The queue is nonempty, so decrement the number of customers in queue. */
        --num_in_q;
        /* Compute the delay of the customer who is beginning service and update the total delay
        accumulator. */
        delay = sim_time - time_arrival[1];
        total_of_delays += delay;
        /* Increment the number of customers delayed, and schedule departure. */
        ++num_custs_delayed;
        time_next_event[2] = sim_time + expon(mean_service);
        /* Move each customer in queue (if any) up one place. */
        for (i = 1; i <= num_in_q; ++i)
            time_arrival[i] = time_arrival[i + 1];
    }
}

```