**Coding challenge**

A Rust program is needed to simulate the operation of a job-processing queue, using FIFO (first in, first out) principles, e.g.

**HEAD TAIL**

|  |  |  |  |
| --- | --- | --- | --- |
| 101  Complaint | 102  Order Revision | 107  Feedback | 110  Order Revision |

**Design notes**

* There is no limit to the length of the job-processing queue.
* Each job created has a unique numeric ID and a bespoke description.
* As noted, the job number must be unique – duplicates must be refused on entry
* Each job is added to the tail of the queue
* Only the chronologically earliest job can be removed, printing its ID and description on removal.
* No further jobs can be removed if the queue is empty (display a suitable error message if this is attempted)
* Minimum menu options: Add job, Remove job, Exit

**Extension tasks**

* Add a menu option to display the number of jobs currently in the queue ;implement this.
* Add a menu option to display the queue contents Head to Tail; implement this.
* Add a menu option to display the queue contents Tail to Head; implement this.
* Add a menu option to insert a “rush” job at a given (valid) position; implement this
* Add a menu option to purge the queue; implement this.