

1. This example demonstrates a single producer and multiple consumers system. The producer generates data at intervals of t seconds and distributes it to k consumers. Each consumer processes the received data as specified. Implement the producer and consumers using **PROCESSES**. The interval t may be as short as 0.1 seconds, and the number of consumers k can scale up to two digits. Both k and t should be specified as input parameters. Each data point generated is a random integer between 0 and 9. Consumers must compute a cumulative sum of all data they receive. The total number of data points to be generated is also an input parameter. The producer must define a special termination character to indicate the end of data transmission. Upon receiving this character, each consumer—identified by a unique ID—should return its cumulative sum and ID to the producer. (**No buffering is permitted for the generated data**; new data must overwrite existing data immediately. The implementation may utilize the **setitimer** system call. Consider implementing a process group and utilizing the **setpgid** system call.)
2. Please implement a **STACK** data structure designed for concurrent access by multiple threads. Additionally, develop a test program to validate its functionality.