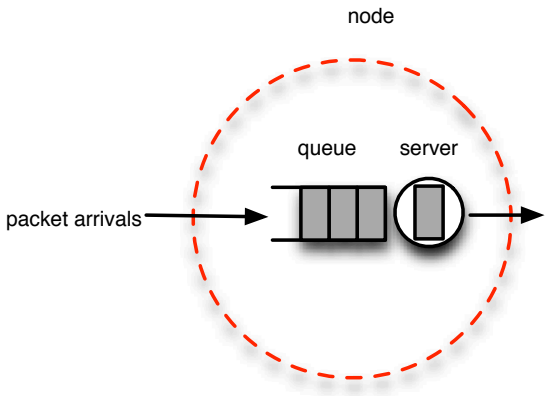
U2267 Probability: Programming Assignment
Due: 12/16 11:59pm

In this programming assignment, we will study how to apply probability theory in practice. There will be two problems. In Problem (1), you are asked to implement a function to generate a random number. By this problem, you can understand how to generate random numbers to meet a statistics property, e.g., CDF. In Problem (2), you are asked to implement a function to simulate a network by employing the function in Problem (1). By this problem, you can understand how to simulate real environment by computer programming. Please implement all the functions in **C code** and compile your code with **GCC**.

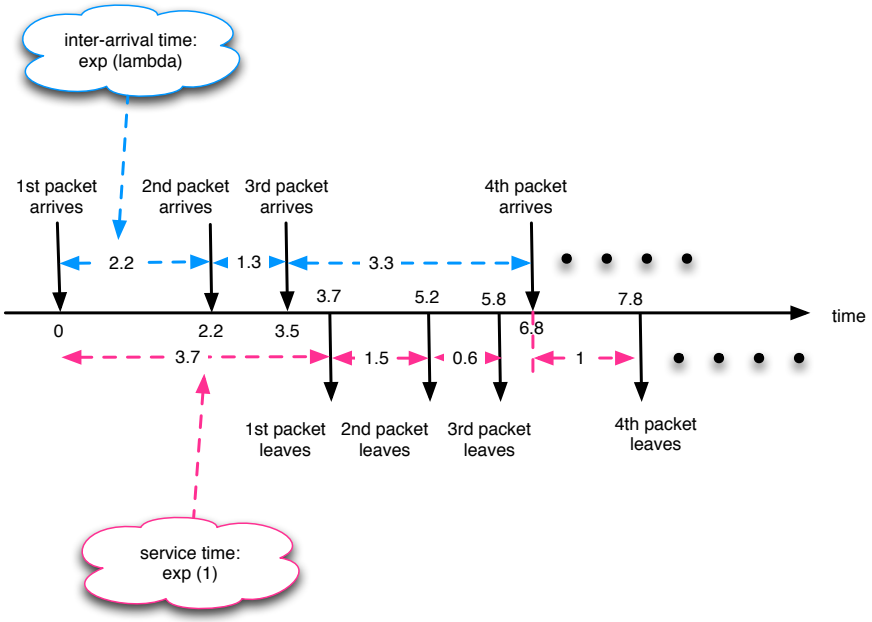
1. Implement the function `double gen_exp(double lambda)` (in `gen_exp_yourID.c` attached along with the document) to generate a random number following the exponential (`lambda`) random variable. Let $\hat{F}(x)$ be the proportion of the random numbers that are no more than x. The main function in `gen_exp_yourID.c` calls the function `gen_exp(double lambda)` with the `lambda` being one to generate 100,000 random numbers, and then calculate $\hat{F}(x)$ for a given x. Moreover, let $F_X(x)$ be the CDF of the exponential (1) random variable. Fill out the form:

x	1	2	3	4	5
$\hat{F}(x)$					
$F_X(x)$					

2. Implement a function `int sim(double lambda)` (in `sim_yourID.c` attached along with the document) to simulate the following network: Consider a node in a network consisting of a server and a queue.



Suppose that packets arrive at the node one-by-one, with the inter-arrival time being the independent exponential (`lambda`) random variable. The server in the node manages packet transmissions and can serve at most one packet for each time, with the service time of a packet being the independent exponential (1) random variable. The queue stores packets that are waiting for service if the server is busy. If the server is available and the queue is not empty, then the packet in the head of the queue will be moved to the server immediately. Use your function `gen_exp(lambda)` in Problem (1) to generate 100,000 random packet arrivals and use your function `gen_exp(1)` to generate their service times. Let N_i, for $i = 1, 2, \cdots, 100,000$, be the number of packets left in the node upon the i-th departure packet. Let T_i, for $i = 1, 2, \cdots, 100,000$ be the amount of time the i-th packet spends in the node, i.e., from its arrival until its departure. The following figure illustrates an example: $N_1 = 2$ (i.e., two packets left upon the first departure), $N_2 = 1$, $N_3 = 0$, $N_4 = 0$, $T_1 = 3.7 - 0 = 3.7$, $T_2 = 5.2 - 2.2 = 3$, $T_3 = 5.8 - 3.5 = 3.3$, $T_4 = 7.8 - 6.8 = 1$.



Let T and N be the sample average of the T_i's and N_i's, respectively, i.e., $T = \frac{\sum_{i=1}^{100,000} T_i}{100,000}$ and $N = \frac{\sum_{i=1}^{100,000} N_i}{100,000}$. The function `sim(double lambda)` prints out the values of N and T for a given `lambda`. Fill out the form:

λ	0.2	0.4	0.6	0.8	1.0	1.2
N						
T						

Please submit the following files to the corresponding folders on the e-learning system.

- (3 points) Submit `gen_exp_yourID.c` (no compress to a rar, zip, ...) to the folder of **code1**. Please replace `yourID` in the file name by **your ID number**.
- (4 points) Submit `sim_yourID.c` (no compress to a rar, zip, ...) to the folder of **code2**. Please replace `yourID` in the file name by **your ID number**.

- (3 points) Submit a `PDF report` (stored as `yourID.pdf` without compression) to the folder of **report** . The report must include
  - high-level ideas of your code,
  - the above two tables,
  - discussion on relationship between N, T, and λ, e.g., $N = \lambda T$, $N = \lambda^2 T, \cdots$,
  - any comment (optional).

Please note:

- I will test your **C code** with **GCC** compiler.
- You are more than welcome to discuss together, but you have to work on the code by yourselves. Do not distribute your code. If your code has similarity over 80% (checked by Turnitin or.....), you will get zero point.
- Again, work by yourself. If you cannot get a perfect version, please be honest and discuss in your report.
- Submit your code and report (without compression) to the correct folder; otherwise, you will get zero point.