

Day #16 Exercise -- SOLUTIONS

Here below is a CSIM model for an M/M/1 queue (stripped of comments, on purpose). Modify the model to keep statistics on how many customers arrive to a system with $j = 0, 1, 2, \dots, 10$ customers in it. At the end of the run, output the percentage of arriving customers that “see” $j = 0, 1, \dots, 10$ customers in the system. The output should look something like this

```
Arrivals that see 0 customers = _____ %
Arrivals that see 1 customers = _____ %
etc.
```

```
#include <stdio.h>
#include "csim.h"

FACILITY Server;

void generate(double lambda, double mu);
void queue1(double mu);

int TotalCount, Count[11];

void sim(void)
{
    double lambda, mu;
    int i;

    create("sim");

    Server = facility("Server");

    lambda = 0.5;
    mu = 1.0;

    TotalCount = 0;
    for (i=0; i<=10; i++)
        Count[i] = 0;

    generate(lambda, mu);
    hold(1000000.0);

    for (i=0; i<=10; i++)
        printf("Arrivals that see %d customers = %f %% \n",
            i, 100.0 * (double) Count[i] / TotalCount);

    report();
}

void generate(double lambda, double mu)
{
    create("generate");

    while(1)
    {
        hold(exponential(1.0 / lambda));
        queue1(mu);
    }
}
```

Here's a run (less the report() output):

```
c:\y>mml_csim
Arrivals that see 0 customers = 49.842833 %
Arrivals that see 1 customers = 24.908835 %
Arrivals that see 2 customers = 12.551973 %
Arrivals that see 3 customers = 6.255717 %
Arrivals that see 4 customers = 3.149526 %
Arrivals that see 5 customers = 1.608813 %
Arrivals that see 6 customers = 0.820183 %
Arrivals that see 7 customers = 0.413786 %
Arrivals that see 8 customers = 0.210288 %
Arrivals that see 9 customers = 0.107041 %
Arrivals that see 10 customers = 0.059512 %
```

```
void queue1(double mu)
{
    int num;

    create("queue1");

    TotalCount++;
    num = qlength(Server) + num_busy(Server);
    if (num <= 10) Count[num]++;

    reserve(Server);
    hold(exponential(1.0 / mu));
    release(Server);
}
```

Here is another solution (using Qtables)

```
#include <stdio.h>
#include "csim.h"

FACILITY Server;
QTABLE QT;

void generate(double lambda, double mu);
void queue1(double mu);

void sim(void)
{
    double lambda, mu;

    create("sim");

    Server = facility("Server");
    QT = qtable("Queue table");
    qtable_histogram(QT, 10, 0, 9);

    lambda = 0.5;
    mu = 1.0;

    generate(lambda, mu);
    hold(1000000.0);

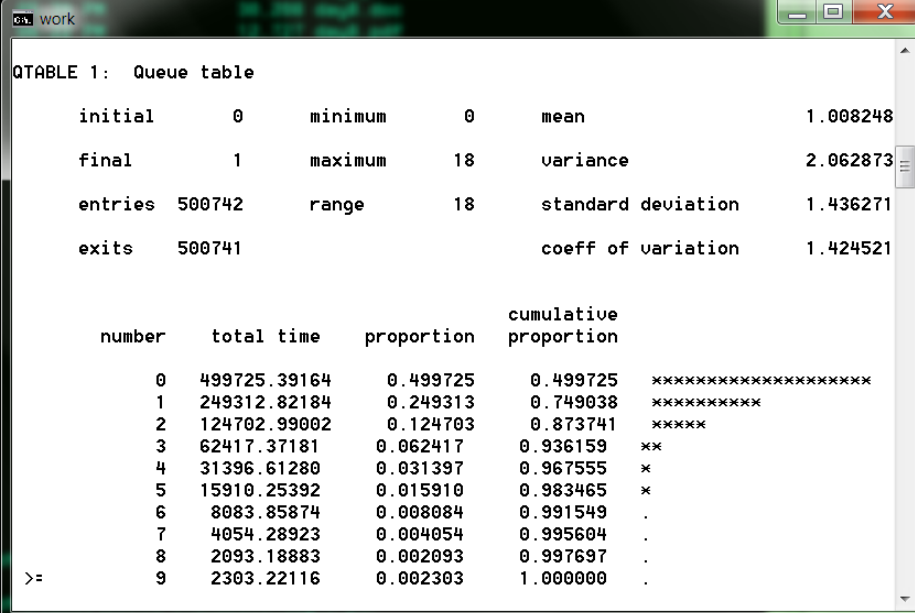
    report();
    report_qtable(QT);
}

void generate(double lambda, double mu)
{
    create("generate");

    while(1)
    {
        hold(exponential(1.0 / lambda));
        queue1(mu);
    }
}

void queue1(double mu)
{
    create("queue1");

    note_entry(QT);
    reserve(Server);
    hold(exponential(1.0 / mu));
    release(Server);
    note_exit(QT);
}
```



QTABLE 1: Queue table

initial	0	minimum	0	mean	1.008248
final	1	maximum	18	variance	2.062873
entries	500742	range	18	standard deviation	1.436271
exits	500741			coeff of variation	1.424521

number	total time	proportion	cumulative proportion		
0	499725.39164	0.499725	0.499725	*****	
1	249312.82184	0.249313	0.749038	*****	
2	124702.99002	0.124703	0.873741	*****	
3	62417.37181	0.062417	0.936159	**	
4	31396.61280	0.031397	0.967555	*	
5	15910.25392	0.015910	0.983465	*	
6	8083.85874	0.008084	0.991549	.	
7	4054.28923	0.004054	0.995604	.	
8	2093.18883	0.002093	0.997697	.	
>=	9	2303.22116	0.002303	1.000000	.