**Pgm #4 Machine Improvement Proposals (40 points + extra credit)**

Server 1 is a bottleneck and does many things. Its step 1 takes an average of 20 tu and its step 2 takes an average of 15 tu. It can only process one widget at a time. While a widget is on step 2, no widgets can be serviced on step 1.

Alternative A. Add a server 2 that only does "step 2" functions. Server 1 will then become a "step 1"-only server. This means widgets go to server 1 (for step 1) and then server 2 for step 2. Different widgets can be processed on both servers at the same time.

Alternative B. Upgrade server 1. The upgrade will reduce the average processing time for step 1 from 20 tu to 15 tu. The step 2 time will not be changed. Server 1 is still processing both steps. While a widget is on step 2, no widgets can be serviced on step 1.

The downtime to reconfigure is expected to be the same for both alternatives so it isn't a factor.

Statistics to produce:

* Number of widgets processed
* Average time in system (from arrival to completion at step 2)
* Average queue time step 1
* Average queue time step 2 (not produced for Alternative B or current)

**Input A (same for alternatives A and current)** - p4ACinput.txt

lWidgetNr iStep1tu iStep2tu iArrivalDelta

%ld %d %d %d

**Input B (for alternative B)** - p4Binput.txt

lWidgetNr iStep1tu iStep2tu iArrivalDelta

%ld %d %d %d

**Points**

**Part A - 40 points**

* Provide a simulation of **alternative A**.
* Include a switch (-v) which has no corresponding argument. If -v is specified, this is verbose mode. This causes your program to print all clock times and events as they happen: Arrival, Enter Queue 1, Leave Queue 1, Seize Server 1, Release Server 1, Enter Queue 2, Leave Queue 2, Seize Server 2, Release Server 2, Exit System. For some of those events, you should print additional information (time in queue, time in system). See the sample output below.
* Show the statistics specified above.
* Run the simulation for 500 tu, not generating new arrivals after time 500, but do allow all of them to complete both servers and exit the system.

**Part B - 15 points (extra credit)**

* Provide simulations of the **current** server processing and independently **alternative B**.
* Include a switch (-v) which has no corresponding argument. If -v is specified, this is verbose mode. This causes your program to print all clock times and events as they happen: arrival, Enter Queue 1, Leave Queue 1, Begin Processing Server 1, Complete Processing Server 1, Exit System
* Provide switches that specify the alternative (-aA for alternative A, -aB for alternative B or -aC for current) to execute.
* When simulating for alternative B and current, make certain you use **both iStep1tu and iStep2tu** (not just iStep1tu).
* Show the statistics specified above.
* Run the simulation for 500 tu, not generating new arrivals after time 500, but do allow all of them to complete and exit the system.
* Provide **a one page paper** which explains which alternative you recommend based on the statistics you provided.

**Sample Output:**

Simulation for Alternative A

Time Widget Event

0 1 Arrived

0 1 Enter Queue 1

0 1 Leave Queue 1, waited 0

0 1 Seized Server 1

13 2 Arrived

13 2 Enter Queue 1

15 1 Released Server 1

15 2 Leave Queue 1, waited 2

15 2 Seized Server 1

15 1 Enter Queue 2

15 1 Leave Queue 2, waited 0

15 1 Seized Server 2

28 1 Released Server 2

28 1 Exit System, in system 28

29 3 Arrived

29 3 Enter Queue 1

46 2 Released Server 1

46 3 Leave Queue 1, waited 17

46 3 Seized Server 1

46 2 Enter Queue 2

46 2 Leave Queue 2, waited 0

46 2 Seized Server 2

48 4 Arrived

48 4 Enter Queue 1

53 5 Arrived

53 5 Enter Queue 1

59 3 Released Server 1

59 4 Leave Queue 1, waited 11

59 4 Seized Server 1

59 3 Enter Queue 2

…

621 Simulation Complete for Alternative A

Average Queue Time for Server 1 = ???.?

Average Queue Time for Server 2 = ???.?

Average Time in System = ???.?