Final project

과제물

학과 : 독어독문학과

학번 : 2013314587

이름 : 안현장

목 차

그림 목차

표 목차

1. Total Simulation Results

1.1. performance table

[표 1] performances of theoretical cases

Architecture	Parameters	Turnaround Time	Maximum Throughput
ef-pl	processing time, $p \cong 7.5$	$p \cong 7.5$	$1/p \cong 0.1333$
ef-mul	processing times, $pl1 \cong pl2 \cong pl3 \cong p \cong 7.5$	$3/\text{throughput} \cong 7.5$	$1/pl1 + 1/pl2 + 1/pl3 \cong 3/p \cong 0.4$
ef-pip	processing times $psm1 \cong psm2 \cong psm3 \cong 3.5 \neq p/3$	$psm1 + psm2 + psm3 \cong p \cong 10.5$	$3/p \cong 0.2857$
ef-dc	processing times, $cp, cm = 3.5$, $psm1 \cong psm2 \cong psm3 \cong 3.5 \neq p/3$	$cp + cm + \frac{p}{3} \cong 10.5$	$1/\max\{p/3, cp, cm\} \cong 0.2857$

[표 2] performances of simulation cases

Architecture	Parameters	Turnaround Time						Maximum Throughput					
		1	2	3	4	5	total	1	2	3	4	5	total
ef-pl	pt	7	6	8	8	8	**7.4**	0.1086	0.1612	0.1098	0.1089	0.1089	0.11984
ef-mul	pt, $pl1 \cong pl2 \cong pl3 \cong p/3$	7	6	8	8	8	**7.4**	0.3333	0.3333	0.3296	0.3296	0.3296	0.33108
ef-pip	pt, $psm1 \cong psm2 \cong psm3 \cong p/3$	6	6	6	6	6	**6**	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
ef-dc	pt, cp, cm, $psm1 \cong psm2 \cong psm3 \cong p/3$,	7	6	6	6	6	**6.2**	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333

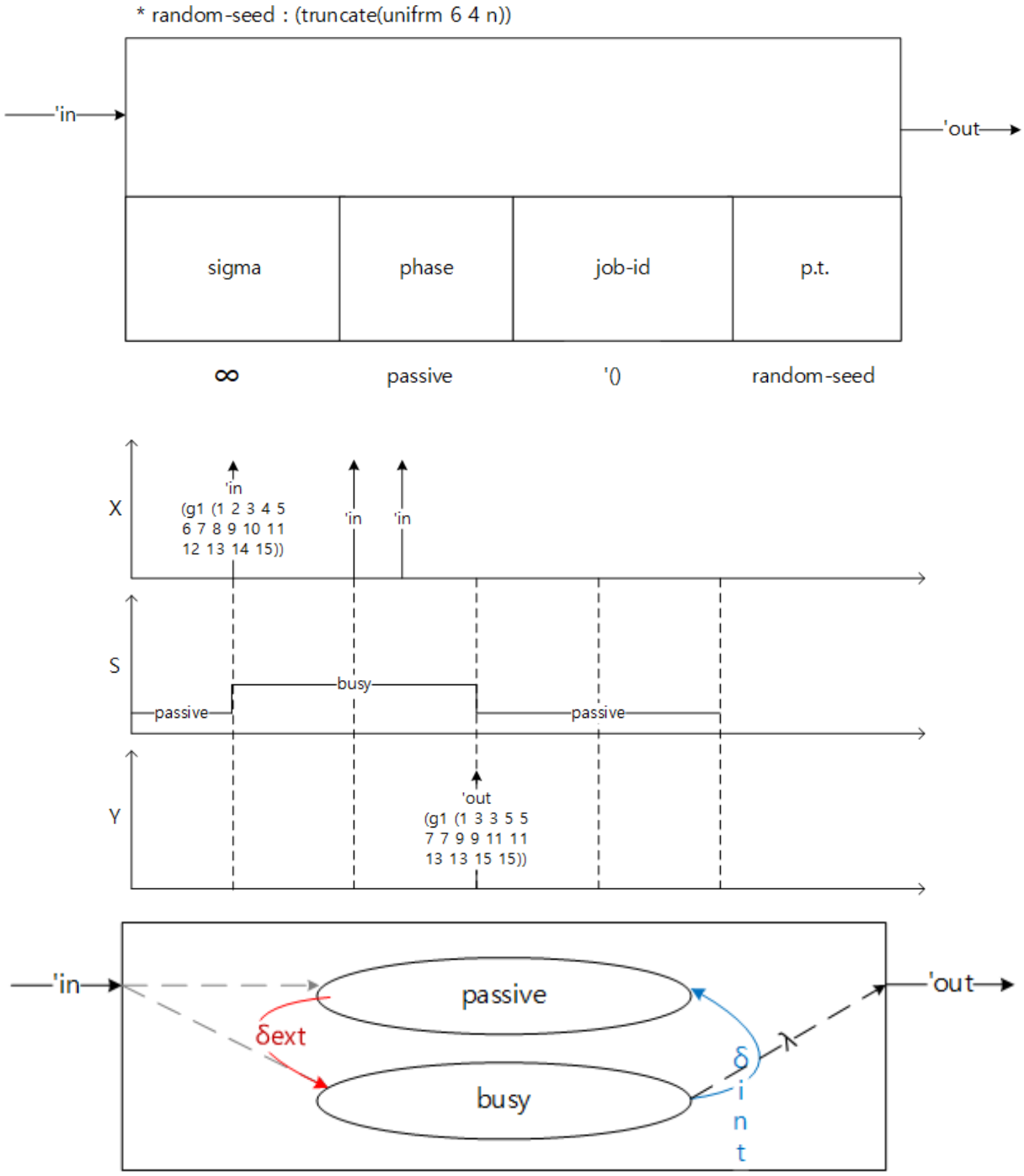
1.2. Discussions

simulation cases 의 결과는 교재에서 참고한 theoretical cases 예측과 비슷하다. 이는 atomic model 만 다를 뿐 architecture 가 동일하기 때문이다. atomic model 로 pl 을 사용하는 Architecture 는 pt 로 6~9 값을 갖고, psm 을 사용하는 모델은 2~5 의 값을 갖는다. (truncate(unifrm))함수로 뽑히는 정수의 확률이 모두 동일하다고 가정했을 때, pl 모델의 pt 기댓값은 7.5 이며, psm 모델의 pt 기댓값은 3.5 이다.

ef-pl 모델과 ef-mul 모델의 경우 예측과 실측이 거의 동일하므로 ef-pip 과 ef-dc 모델의 예측 오차에 보다 집중한다. 예측오차의 이유는 psm 모델의 pt 실제값이 기댓값보다 낮았기 때문으로 추측되며 이는 오히려 psm 모델의 pt 값이 pl 모델의 pt 의 삼분의 일에 가까운 값으로 생성되었다는 사실도 추측할 수 있다.

2. atomic models

2.1. pl

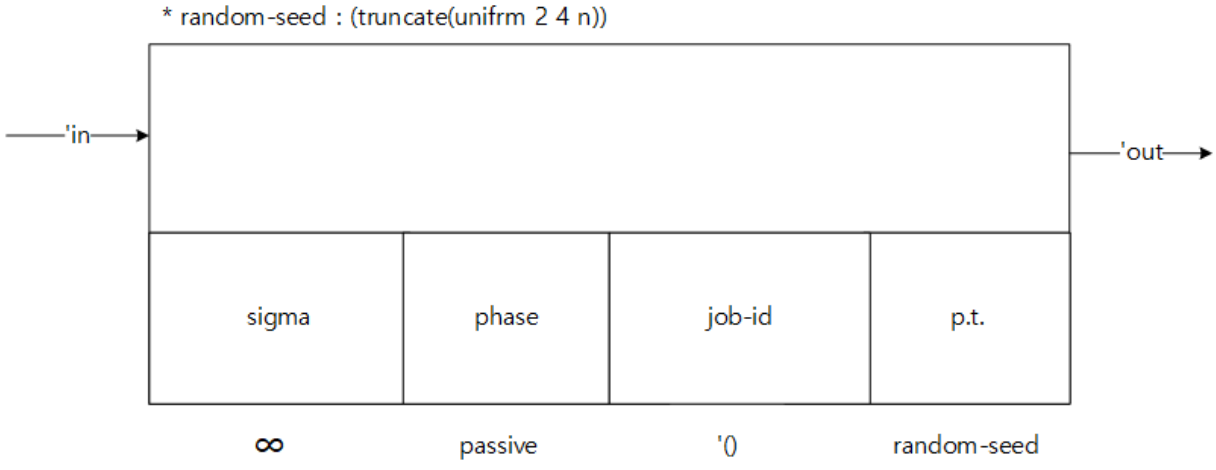


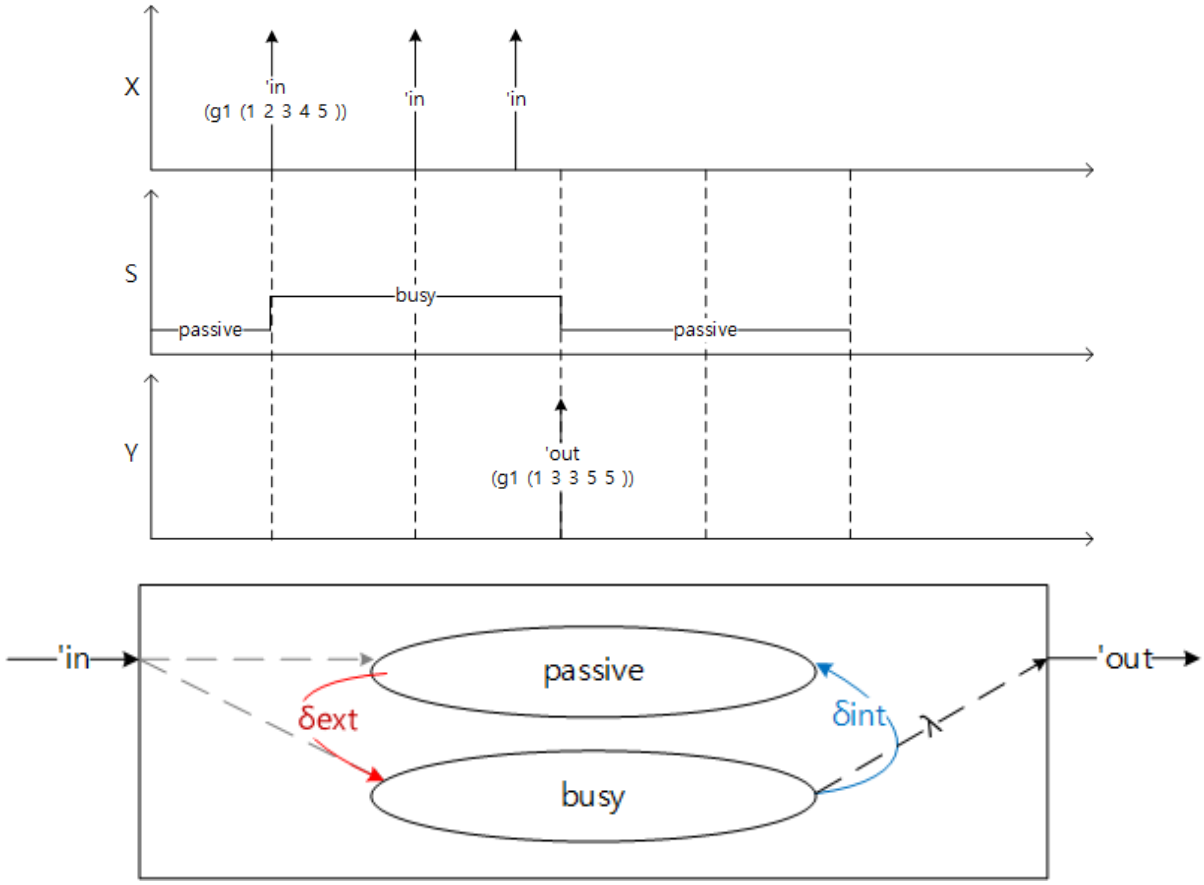
[그림 1] pl model

```
OK
[13] (load "mbase/pl.m")

Model of type atomic-models with name PL made.
Processor of type simulators with name S:PL made.
OK
[14] (send pl inject 'in '(g1 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)))
state s =
                                                    state s = (9 BUSY (G1
(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)) 9)state s = ()()
[15] (send pl inject 'in '(g2 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)))
state s =

state s = (9 BUSY (G1 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)) 9)state s = ()()
[16] (send pl output?)
output y =      output y = OUT (G1 (1 3 3 5 5 7 7 9 9
11 11 13 13 15 15))#(((|#!STRUCTURE| . CONTENT)) OUT (G1 (1 3 3 5 5 7 7 9 9 11 11 13
13 15 15)))
[17] (send pl int-transition)
state s =

state s = (INF PASSIVE (G1 (1 3 3 5 5 7 7 9 9 11 11 13 13 15 15)) 9)state s = ()()
[18] (transcript-off)
```

[그림 2] pl test file

2.2. psm



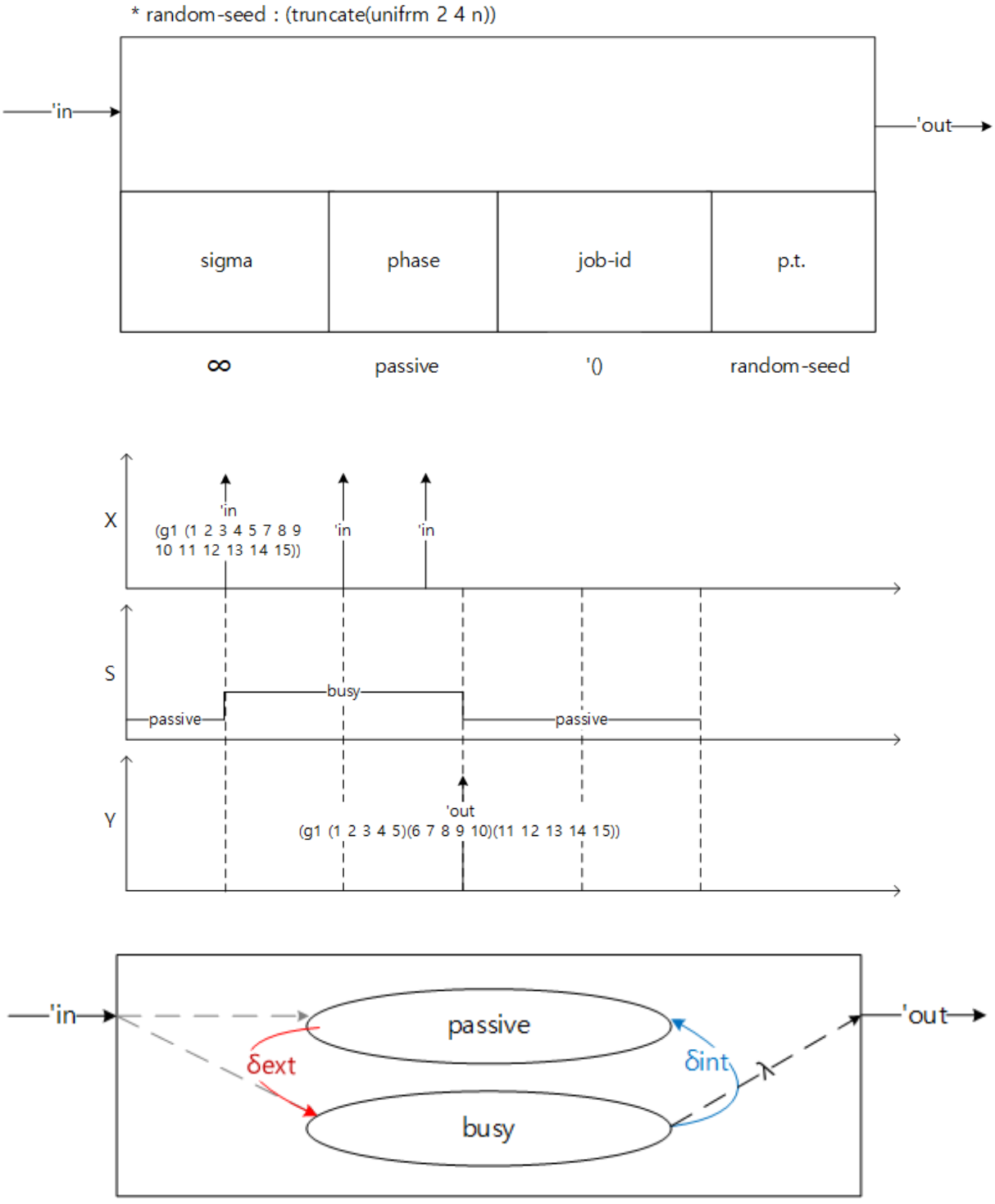


[그림 3] psm model

```
OK
[20] (load "mbase/psm.m")

Model of type atomic-models with name PSM made.
Processor of type simulators with name S:PSM made.
OK
[21] (send psm inject 'in '(g1 (1 2 3 4 5)))
state s =
                                                    state s = (2 BUSY (G1
(1 2 3 4 5)) 2)state s = ()()
[22] (send psm inject 'in '(g2 (1 2 3 4 5)))
state s =
                                                        state s = (2
BUSY (G1 (1 2 3 4 5)) 2)state s = ()()
[23] (send psm output?)
output y =      output y = OUT (G1 (1 3 3 5 5))#((((|#!STRUCTURE| . CONTENT)) OUT (G1 (1
3 3 5 5)))
[24] (send psm int-transition)
state s =
                                                        state s =
(INF PASSIVE (G1 (1 3 3 5 5)) 2)state s = ()()
[25] (transcipt-off)

[VM ERROR encountered!] Variable not defined in current environment
TRANSCIPT-OFF

[Inspect] Quit
[26] (transcript-off)
```

[그림 4] psm test file

2.3. pd

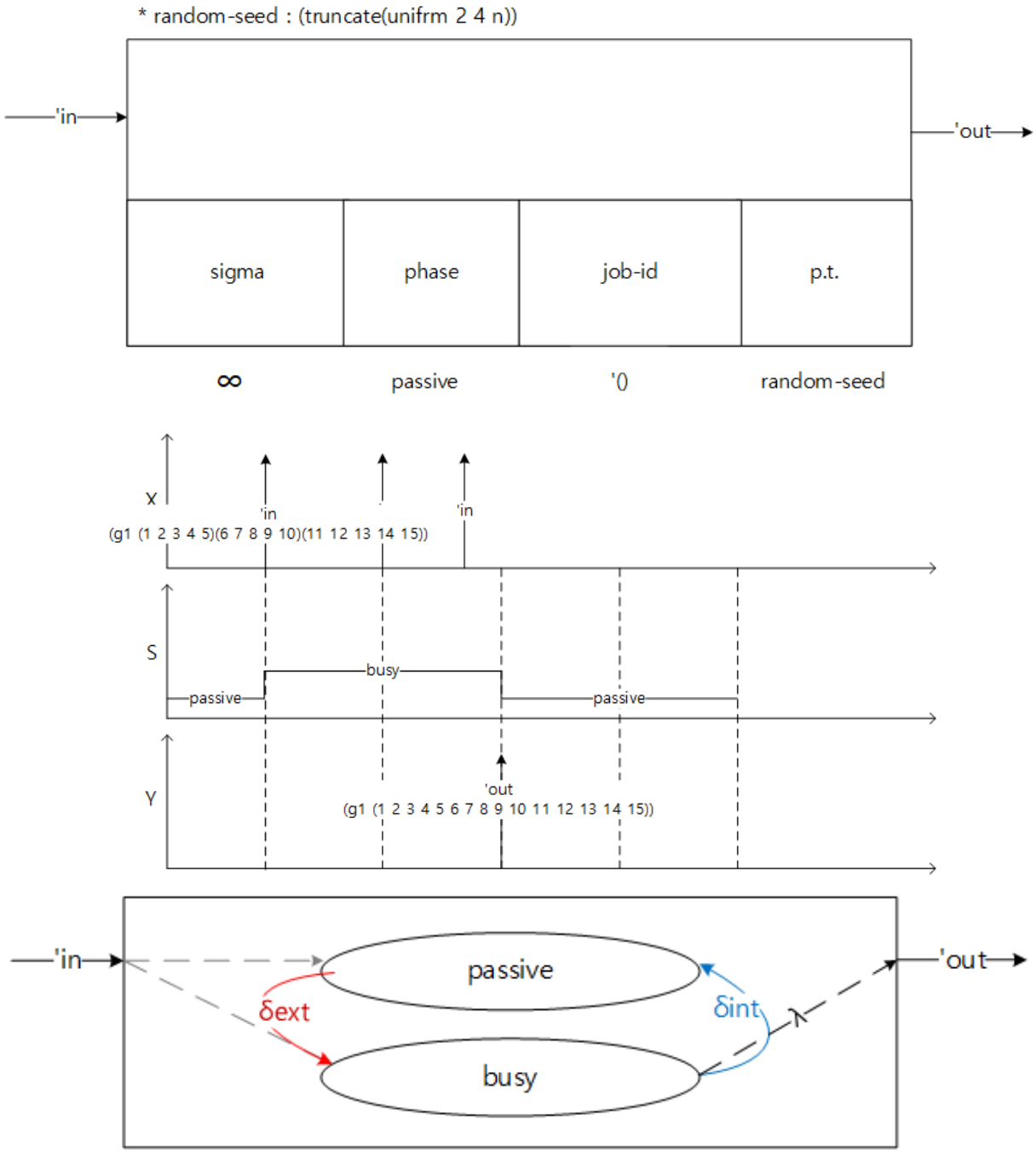


[그림 5] pd model

```
OK
[28] (load "mbase/pd.m")

Model of type atomic-models with name PD made.
Processor of type simulators with name S:PD made.
OK
[29] (send pd inject 'in '(g1 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)))
state s =
                                                              state s = (3 BUSY (G1
(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)) 3)state s = ()()
[30] (send pd inject 'in '(g2 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)))
state s =

state s = (3 BUSY (G1 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)) 3)state s = ()()
[31] (send pd output?)
output y =      output y = OUT (G1 (1 2 3 4 5) (6 7 8
9 10) (11 12 13 14 15))#((((|#!STRUCTURE| . CONTENT)) OUT (G1 (1 2 3 4 5) (6 7 8 9 10)
(11 12 13 14 15)))
[32] (send pd int-transition)
state s =

state s = (INF PASSIVE (G1 (1 2 3 4 5) (6 7 8 9 10) (11 12 13 14 15)) 3)state s = ()()
[33] (transcript-off)
```

[그림 6] pd test file

2.4. pc



[그림 7] pc model

```
OK
[39] (load "mbase/pc.m")

Model of type atomic-models with name PC made.
Processor of type simulators with name S:PC made.
OK
[40] (send pc inject 'in '(g1 (1 2 3 4 5)(6 7 8 9 10)(11 12 13 14 15)))
state s =
                                                          state s = (3 BUSY (G1
(1 2 3 4 5) (6 7 8 9 10) (11 12 13 14 15)) 3)state s = ()()
[41] (send pc inject 'in '(g2 (1 2 3 4 5)(6 7 8 9 10)(11 12 13 14 15)))
state s =

state s = (3 BUSY (G1 (1 2 3 4 5) (6 7 8 9 10) (11 12 13 14 15)) 3)state s = ()()
[42] (send pc output?)
output y =      output y = OUT (G1 (1 2 3 4 5 6 7 8 9
10 11 12 13 14 15))#((((|#!STRUCTURE| . CONTENT)) OUT (G1 (1 2 3 4 5 6 7 8 9 10 11 12
13 14 15)))
[43] (send pc int-transition)
state s =

state s = (INF PASSIVE (G1 (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)) 3)state s = ()()
[44] (transcript-off)
```

[그림 8] pc test file

3. log files

3.1. log files

3.1.1. ef-pl

```
The arrived list: ((G41 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G40 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G39 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G38 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G37 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G36 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G35 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G34 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G33 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G32 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G31 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G30 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G29 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G28 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G27 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G26 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G25 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G24 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G23 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G22 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G21 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G20 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G19 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G18 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G17 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G16 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G15 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G14 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G13 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G12 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G11 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G10 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G9 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G8 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G38 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G35 (3
67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G32 (95 97 13 1 39 23 25 53 71 57 25
3 79 7 21)) (G29 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G26 (33 23 31 97
47 65 29 13 3 49 77 15 69 91 87)) (G23 (17 43 97 25 77 77 23 31 71 27 5 37 31 63
69)) (G20 (45 51 81 1 99 81 27 63 17 67 99 45 41 7 7)) (G17 (27 19 51 83 1 11 79
87 41 15 29 75 75 57 23)) (G14 (9 93 63 53 43 75 29 37 87 11 3 79 69 83 1)) (G11
(3 83 93 17 7 97 7 63 57 73 9 7 95 83 45)) (G8 (47 79 95 75 49 75 19 87 41 81 35
15 17 61 83)))
Avg. turnaround time: 7
ThruPut: 0.108695652173913
```

```
The arrived list: ((G407 (22 6 20 32 50 82 39 86 43 70 53 34 20 1 37) 99) (G406
(1 59 23 40 21 99 4 18 59 92 26 32 67 76 58) 96) (G405 (53 50 14 15 57 79 93 16
2 25 87 96 98 12 3) 93) (G404 (33 37 74 81 87 70 8 8 32 40 28 35 88 18 12) 90) (
G403 (84 26 32 16 74 97 10 6 41 43 57 23 71 40 87) 87) (G402 (67 80 64 62 66 97
54 16 78 83 8 50 11 52 59) 84) (G401 (92 10 67 14 85 64 21 98 98 24 90 85 20 23
35) 81) (G400 (63 68 59 7 3 93 17 32 59 55 7 85 54 12 4) 78) (G399 (35 52 63 26
51 76 47 85 48 33 53 47 67 7 31) 75) (G398 (23 62 26 26 84 0 39 83 64 50 8 11 70
4 76) 72) (G397 (73 66 98 33 22 99 91 79 34 25 81 79 77 1 48) 69) (G396 (75 21
63 44 9 23 22 44 6 98 35 29 32 22 99) 66) (G395 (9 67 55 61 14 34 78 17 91 33 24
59 31 86 44) 63) (G394 (2 91 66 0 96 76 90 2 57 80 25 26 45 52 65) 60) (G393 (14
52 80 29 84 20 2 23 55 93 57 86 73 29 39) 57) (G392 (7 65 62 23 40 34 68 39 66
91 0 99 3 69 63) 54) (G391 (50 66 73 8 52 38 59 1 82 43 30 96 5 72 39) 51) (G390
(25 87 57 61 82 28 69 6 85 44 42 70 73 99 47) 48) (G389 (28 22 9 97 1 24 90 91
98 59 41 4 47 53 88) 45) (G388 (45 21 32 35 60 98 19 3 17 50 85 4 55 88 35) 42)
(G387 (87 17 23 28 57 37 15 53 76 9 42 1 1 48 56) 39) (G386 (27 39 24 2 14 85 59
49 22 64 1 38 67 41 78) 36) (G385 (35 19 32 14 19 36 12 87 88 69 58 46 35 50 4)
33) (G384 (58 28 39 79 0 94 3 57 87 78 69 43 71 81 4) 30) (G383 (72 48 27 31 40
97 56 76 60 24 60 25 45 54 98) 27) (G382 (78 89 15 65 26 32 11 10 91 0 6 48 10
41 5) 24) (G381 (58 44 78 50 87 53 46 40 8 75 49 22 51 23 62) 21) (G380 (84 47
54 64 23 2 89 25 19 37 94 30 13 89 2) 18) (G379 (12 98 7 75 15 17 14 77 73 21 50
46 85 39 21) 15) (G378 (44 94 97 8 57 57 87 33 95 70 1 14 64 46 5) 12) (G377 (55
56 42 41 59 42 57 89 32 87 9 16 41 53 8) 9) (G376 (98 41 17 4 28 97 58 13 99 25
8 31 81 53 93) 6) (G375 (15 85 78 30 39 83 57 14 98 6 5 0 61 83 90) 3) (G374 (11
33 23 5 8 27 95 77 87 60 5 6 84 81 74) 0))
The solved list:  ((G404 (33 37 75 81 87 71 9 9 33 41 29 35 89 19 13)) (G402 (67
81 65 63 67 97 55 17 79 83 9 51 11 53 59)) (G400 (63 69 59 7 3 93 17 33 59 55 7
85 55 13 5)) (G398 (23 63 27 27 85 1 39 83 65 51 9 11 71 5 77)) (G396 (75 21 63
45 9 23 23 45 7 99 35 29 33 23 99)) (G394 (3 91 67 1 97 77 91 3 57 81 25 27 45
53 65)) (G392 (7 65 63 23 41 35 69 39 67 91 1 99 3 69 63)) (G390 (25 87 57 61 83
29 69 7 85 45 43 71 73 99 47)) (G388 (45 21 33 35 61 99 19 3 17 51 85 5 55 89 35
)) (G386 (27 39 25 3 15 85 59 49 23 65 1 39 67 41 79)) (G384 (59 29 39 79 1 95 3
57 87 79 69 43 71 81 5)) (G382 (79 89 15 65 27 33 11 11 91 1 7 49 11 41 5)) (
G380 (85 47 55 65 23 3 89 25 19 37 95 31 13 89 3)) (G378 (45 95 97 9 57 57 87 33
95 71 1 15 65 47 5)) (G376 (99 41 17 5 29 97 59 13 99 25 9 31 81 53 93)) (G374 (
11 33 23 5 9 27 95 77 87 61 5 7 85 81 75)))
Avg. turnaround time: 6
ThruPut: 0.161290322580645
```

```
The arrived list: ((G41 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G40 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G39 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G38 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G37 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G36 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G35 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G34 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G33 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G32 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G31 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G30 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G29 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G28 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G27 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G26 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G25 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G24 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G23 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G22 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G21 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G20 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G19 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G18 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G17 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G16 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G15 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G14 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G13 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G12 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G11 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G10 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G9 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G8 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G38 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G35 (3
67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G32 (95 97 13 1 39 23 25 53 71 57 25
3 79 7 21)) (G29 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G26 (33 23 31 97
47 65 29 13 3 49 77 15 69 91 87)) (G23 (17 43 97 25 77 77 23 31 71 27 5 37 31 63
69)) (G20 (45 51 81 1 99 81 27 63 17 67 99 45 41 7 7)) (G17 (27 19 51 83 1 11 79
87 41 15 29 75 75 57 23)) (G14 (9 93 63 53 43 75 29 37 87 11 3 79 69 83 1)) (G11
(3 83 93 17 7 97 7 63 57 73 9 7 95 83 45)) (G8 (47 79 95 75 49 75 19 87 41 81 35
15 17 61 83)))
Avg. turnaround time: 8
ThruPut: 0.10989010989011
```

```
The arrived list: ((G41 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G40 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G39 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G38 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G37 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G36 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G35 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G34 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G33 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G32 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G31 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G30 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G29 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G28 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G27 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G26 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G25 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G24 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G23 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G22 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G21 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G20 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G19 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G18 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G17 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G16 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G15 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G14 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G13 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G12 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G11 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G10 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G9 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G8 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G38 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G35 (3
67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G32 (95 97 13 1 39 23 25 53 71 57 25
3 79 7 21)) (G29 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G26 (33 23 31 97
47 65 29 13 3 49 77 15 69 91 87)) (G23 (17 43 97 25 77 77 23 31 71 27 5 37 31 63
69)) (G20 (45 51 81 1 99 81 27 63 17 67 99 45 41 7 7)) (G17 (27 19 51 83 1 11 79
87 41 15 29 75 75 57 23)) (G14 (9 93 63 53 43 75 29 37 87 11 3 79 69 83 1)) (G11
(3 83 93 17 7 97 7 63 57 73 9 7 95 83 45)) (G8 (47 79 95 75 49 75 19 87 41 81 35
15 17 61 83)))
Avg. turnaround time: 8
ThruPut: 0.10989010989011
```


```
The arrived list: ((G41 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G40 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G39 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G38 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G37 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G36 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G35 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G34 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G33 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G32 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G31 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G30 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G29 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G28 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G27 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G26 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G25 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G24 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G23 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G22 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G21 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G20 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G19 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G18 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G17 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G16 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G15 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G14 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G13 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G12 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G11 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G10 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G9 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G8 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G38 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G35 (3
67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G32 (95 97 13 1 39 23 25 53 71 57 25
3 79 7 21)) (G29 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G26 (33 23 31 97
47 65 29 13 3 49 77 15 69 91 87)) (G23 (17 43 97 25 77 77 23 31 71 27 5 37 31 63
69)) (G20 (45 51 81 1 99 81 27 63 17 67 99 45 41 7 7)) (G17 (27 19 51 83 1 11 79
87 41 15 29 75 75 57 23)) (G14 (9 93 63 53 43 75 29 37 87 11 3 79 69 83 1)) (G11
(3 83 93 17 7 97 7 63 57 73 9 7 95 83 45)) (G8 (47 79 95 75 49 75 19 87 41 81 35
15 17 61 83)))
Avg. turnaround time: 8
ThruPut: 0.10989010989011
```

3.1.2. ef-mul

```
The arrived list: ((G91 (86 44 6 71 40 24 41 8 86 29 5 34 47 8 92) 99) (G90 (49
57 9 68 48 97 6 2 47 61 60 79 10 20 5) 96) (G89 (53 41 32 86 40 9 91 8 81 30 4
74 52 60 35) 93) (G88 (89 66 74 75 53 86 96 77 22 20 79 26 36 66 60) 90) (G87 (
53 22 29 63 89 40 63 98 86 25 51 22 12 65 21) 87) (G86 (0 19 94 22 74 32 13 69
39 30 6 45 91 87 59) 84) (G85 (4 30 63 20 55 68 2 22 55 3 41 41 67 53 84) 81) (
G84 (15 16 78 28 21 40 49 86 13 63 17 25 47 84 76) 78) (G83 (75 5 38 40 27 81 89
11 46 15 93 3 52 5 80) 75) (G82 (56 91 47 83 10 41 69 49 74 57 66 45 63 53 39)
72) (G81 (60 19 15 21 47 39 84 76 40 49 26 49 15 23 72) 69) (G80 (10 68 13 45 71
24 26 78 16 19 97 99 90 19 9) 66) (G79 (3 32 27 33 11 39 95 29 34 42 77 63 36 92
86) 63) (G78 (50 20 72 40 77 31 24 40 2 49 86 9 92 80 19) 60) (G77 (54 66 85 75
64 15 83 35 40 64 17 82 53 85 89) 57) (G76 (89 11 7 56 19 90 56 31 32 21 94 91
49 75 41) 54) (G75 (34 88 41 66 22 17 1 85 27 43 26 73 33 78 76) 51) (G74 (25 69
12 94 83 23 32 98 79 7 70 67 92 13 98) 48) (G73 (21 67 23 40 93 55 58 41 7 52 48
13 22 8 84) 45) (G72 (0 40 96 74 85 15 23 8 72 43 3 51 85 77 5) 42) (G71 (47 77
84 52 65 30 35 48 8 6 81 61 14 10 44) 39) (G70 (86 66 75 76 23 10 8 78 64 45 95
64 50 55 7) 36) (G69 (31 4 92 40 83 74 36 95 18 85 96 78 45 33 12) 33) (G68 (52
96 73 70 52 95 60 77 87 71 68 44 73 23 76) 30) (G67 (97 51 73 3 9 79 16 98 62 96
76 64 57 89 55) 27) (G66 (88 18 95 75 39 9 98 13 82 35 56 53 38 90 73) 24) (G65
(67 64 22 72 28 40 74 45 46 75 12 91 51 9 53) 21) (G64 (99 26 15 44 19 7 67 62
24 60 18 12 63 80 56) 18) (G63 (43 69 83 36 34 63 44 10 17 68 96 67 79 9 83) 15)
(G62 (94 75 7 21 14 72 78 73 56 78 23 20 79 32 64) 12) (G61 (9 16 76 23 58 77 15
73 67 21 33 13 79 89 48) 9) (G60 (94 69 92 99 82 50 30 85 57 37 85 14 33 49 41)
6) (G59 (93 95 50 23 24 19 98 27 43 42 76 52 18 40 76) 3) (G58 (78 23 23 26 49
76 21 59 34 26 3 74 15 80 84) 0))
The solved list:  ((G89 (53 41 33 87 41 9 91 9 81 31 5 75 53 61 35)) (G88 (89 67
75 75 53 87 97 77 23 21 79 27 37 67 61)) (G87 (53 23 29 63 89 41 63 99 87 25 51
23 13 65 21)) (G86 (1 19 95 23 75 33 13 69 39 31 7 45 91 87 59)) (G85 (5 31 63
21 55 69 3 23 55 3 41 41 67 53 85)) (G84 (15 17 79 29 21 41 49 87 13 63 17 25 47
85 77)) (G83 (75 5 39 41 27 81 89 11 47 15 93 3 53 5 81)) (G82 (57 91 47 83 11
41 69 49 75 57 67 45 63 53 39)) (G81 (61 19 15 21 47 39 85 77 41 49 27 49 15 23
73)) (G80 (11 69 13 45 71 25 27 79 17 19 97 99 91 19 9)) (G79 (3 33 27 33 11 39
95 29 35 43 77 63 37 93 87)) (G78 (51 21 73 41 77 31 25 41 3 49 87 9 93 81 19))
(G77 (55 67 85 75 65 15 83 35 41 65 17 83 53 85 89)) (G76 (89 11 7 57 19 91 57
31 33 21 95 91 49 75 41)) (G75 (35 89 41 67 23 17 1 85 27 43 27 73 33 79 77)) (
G74 (25 69 13 95 83 23 33 99 79 7 71 67 93 13 99)) (G73 (21 67 23 41 93 55 59 41
7 53 49 13 23 9 85)) (G72 (1 41 97 75 85 15 23 9 73 43 3 51 85 77 5)) (G71 (47
77 85 53 65 31 35 49 9 7 81 61 15 11 45)) (G70 (87 67 75 77 23 11 9 79 65 45 95
65 51 55 7)) (G69 (31 5 93 41 83 75 37 95 19 85 97 79 45 33 13)) (G68 (53 97 73
71 53 95 61 77 87 71 69 45 73 23 77)) (G67 (97 51 73 3 9 79 17 99 63 97 77 65 57
89 55)) (G66 (89 19 95 75 39 9 99 13 83 35 57 53 39 91 73)) (G65 (67 65 23 73 29
41 75 45 47 75 13 91 51 9 53)) (G64 (99 27 15 45 19 7 67 63 25 61 19 13 63 81 57
)) (G63 (43 69 83 37 35 63 45 11 17 69 97 67 79 9 83)) (G62 (95 75 7 21 15 73 79
73 57 79 23 21 79 33 65)) (G61 (9 17 77 23 59 77 15 73 67 21 33 13 79 89 49)) (
G60 (95 69 93 99 83 51 31 85 57 37 85 15 33 49 41)) (G59 (93 95 51 23 25 19 99
27 43 43 77 53 19 41 77)) (G58 (79 23 23 27 49 77 21 59 35 27 3 75 15 81 85)))
Avg. turnaround time: 7
ThruPut: 0.333333333333333
```

```
The arrived list: ((G49 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G48 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G47 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G46 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G45 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G44 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G43 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G42 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G41 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G40 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G39 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G38 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G37 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G36 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G35 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G34 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G33 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G32 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G31 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G30 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G29 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G28 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G27 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G26 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G25 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G24 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G23 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G22 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G21 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G20 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G19 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G18 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G17 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G16 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G47 (39 91 13 79 49 27 89 53 11 83 11 35 43 97 23)) (G46 (19
15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G45 (49 75 11 47 37 1 21 31 19 5 69
61 75 39 29)) (G44 (49 59 11 85 97 67 61 29 83 29 9 63 5 57 31)) (G43 (3 67 15
93 1 25 85 27 47 61 69 21 93 73 21)) (G42 (45 21 23 71 73 39 25 99 91 11 85 3 19
97 31)) (G41 (37 11 75 53 15 79 47 37 25 99 71 5 93 73 45)) (G40 (95 97 13 1 39
23 25 53 71 57 25 3 79 7 21)) (G39 (11 77 37 43 1 21 69 25 79 37 95 41 83 43 5))
(G38 (75 47 59 53 71 37 73 21 1 91 35 87 49 99 77)) (G37 (79 15 63 13 5 1 47 21
15 55 69 37 63 81 53)) (G36 (99 39 71 17 61 61 61 99 1 89 91 27 95 41 45)) (G35
(35 69 23 61 15 21 59 13 87 67 9 9 67 71 99)) (G34 (33 23 31 97 47 65 29 13 3 49
77 15 69 91 87)) (G33 (15 73 1 37 77 25 91 7 3 9 97 3 65 19 29)) (G32 (51 19 43
19 13 73 73 85 33 95 47 13 75 57 47)) (G31 (17 43 97 25 77 77 23 31 71 27 5 37
31 63 69)) (G30 (41 9 95 37 51 45 57 79 83 65 75 1 95 77 95)) (G29 (15 21 27 7
85 29 21 93 99 69 39 77 23 19 47)) (G28 (45 51 81 1 99 81 27 63 17 67 99 45 41 7
7)) (G27 (73 67 21 43 33 11 13 49 5 95 95 31 17 5 71)) (G26 (49 59 27 11 53 63 7
21 65 9 15 13 97 91 29)) (G25 (27 19 51 83 1 11 79 87 41 15 29 75 75 57 23)) (
G24 (53 41 95 57 53 51 65 43 51 29 11 23 3 23 3)) (G23 (29 59 3 79 15 93 95 13
95 19 61 3 1 45 93)) (G22 (9 93 63 53 43 75 29 37 87 11 3 79 69 83 1)) (G21 (1
39 35 31 93 27 23 47 41 27 31 21 7 89 19)) (G20 (47 69 11 13 95 49 81 53 19 33
67 73 55 7 85)) (G19 (3 83 93 17 7 97 7 63 57 73 9 7 95 83 45)) (G18 (81 65 11
51 93 51 89 9 81 25 83 83 53 5 37)) (G17 (97 41 37 67 9 85 37 95 51 55 29 97 89
73 27)) (G16 (47 79 95 75 49 75 19 87 41 81 35 15 17 61 83)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G91 (86 44 6 71 40 24 41 8 86 29 5 34 47 8 92) 99) (G90 (49
57 9 68 48 97 6 2 47 61 60 79 10 20 5) 96) (G89 (53 41 32 86 40 9 91 8 81 30 4
74 52 60 35) 93) (G88 (89 66 74 75 53 86 96 77 22 20 79 26 36 66 60) 90) (G87 (
53 22 29 63 89 40 63 98 86 25 51 22 12 65 21) 87) (G86 (0 19 94 22 74 32 13 69
39 30 6 45 91 87 59) 84) (G85 (4 30 63 20 55 68 2 22 55 3 41 41 67 53 84) 81) (
G84 (15 16 78 28 21 40 49 86 13 63 17 25 47 84 76) 78) (G83 (75 5 38 40 27 81 89
11 46 15 93 3 52 5 80) 75) (G82 (56 91 47 83 10 41 69 49 74 57 66 45 63 53 39)
72) (G81 (60 19 15 21 47 39 84 76 40 49 26 49 15 23 72) 69) (G80 (10 68 13 45 71
24 26 78 16 19 97 99 90 19 9) 66) (G79 (3 32 27 33 11 39 95 29 34 42 77 63 36 92
86) 63) (G78 (50 20 72 40 77 31 24 40 2 49 86 9 92 80 19) 60) (G77 (54 66 85 75
64 15 83 35 40 64 17 82 53 85 89) 57) (G76 (89 11 7 56 19 90 56 31 32 21 94 91
49 75 41) 54) (G75 (34 88 41 66 22 17 1 85 27 43 26 73 33 78 76) 51) (G74 (25 69
12 94 83 23 32 98 79 7 70 67 92 13 98) 48) (G73 (21 67 23 40 93 55 58 41 7 52 48
13 22 8 84) 45) (G72 (0 40 96 74 85 15 23 8 72 43 3 51 85 77 5) 42) (G71 (47 77
84 52 65 30 35 48 8 6 81 61 14 10 44) 39) (G70 (86 66 75 76 23 10 8 78 64 45 95
64 50 55 7) 36) (G69 (31 4 92 40 83 74 36 95 18 85 96 78 45 33 12) 33) (G68 (52
96 73 70 52 95 60 77 87 71 68 44 73 23 76) 30) (G67 (97 51 73 3 9 79 16 98 62 96
76 64 57 89 55) 27) (G66 (88 18 95 75 39 9 98 13 82 35 56 53 38 90 73) 24) (G65
(67 64 22 72 28 40 74 45 46 75 12 91 51 9 53) 21) (G64 (99 26 15 44 19 7 67 62
24 60 18 12 63 80 56) 18) (G63 (43 69 83 36 34 63 44 10 17 68 96 67 79 9 83) 15)
(G62 (94 75 7 21 14 72 78 73 56 78 23 20 79 32 64) 12) (G61 (9 16 76 23 58 77 15
73 67 21 33 13 79 89 48) 9) (G60 (94 69 92 99 82 50 30 85 57 37 85 14 33 49 41)
6) (G59 (93 95 50 23 24 19 98 27 43 42 76 52 18 40 76) 3) (G58 (78 23 23 26 49
76 21 59 34 26 3 74 15 80 84) 0))
The solved list:  ((G88 (89 67 75 75 53 87 97 77 23 21 79 27 37 67 61)) (G87 (53
23 29 63 89 41 63 99 87 25 51 23 13 65 21)) (G86 (1 19 95 23 75 33 13 69 39 31 7
45 91 87 59)) (G85 (5 31 63 21 55 69 3 23 55 3 41 41 67 53 85)) (G84 (15 17 79
29 21 41 49 87 13 63 17 25 47 85 77)) (G83 (75 5 39 41 27 81 89 11 47 15 93 3 53
5 81)) (G82 (57 91 47 83 11 41 69 49 75 57 67 45 63 53 39)) (G81 (61 19 15 21 47
39 85 77 41 49 27 49 15 23 73)) (G80 (11 69 13 45 71 25 27 79 17 19 97 99 91 19
9)) (G79 (3 33 27 33 11 39 95 29 35 43 77 63 37 93 87)) (G78 (51 21 73 41 77 31
25 41 3 49 87 9 93 81 19)) (G77 (55 67 85 75 65 15 83 35 41 65 17 83 53 85 89))
(G76 (89 11 7 57 19 91 57 31 33 21 95 91 49 75 41)) (G75 (35 89 41 67 23 17 1 85
27 43 27 73 33 79 77)) (G74 (25 69 13 95 83 23 33 99 79 7 71 67 93 13 99)) (G73
(21 67 23 41 93 55 59 41 7 53 49 13 23 9 85)) (G72 (1 41 97 75 85 15 23 9 73 43
3 51 85 77 5)) (G71 (47 77 85 53 65 31 35 49 9 7 81 61 15 11 45)) (G70 (87 67 75
77 23 11 9 79 65 45 95 65 51 55 7)) (G69 (31 5 93 41 83 75 37 95 19 85 97 79 45
33 13)) (G68 (53 97 73 71 53 95 61 77 87 71 69 45 73 23 77)) (G67 (97 51 73 3 9
79 17 99 63 97 77 65 57 89 55)) (G66 (89 19 95 75 39 9 99 13 83 35 57 53 39 91
73)) (G65 (67 65 23 73 29 41 75 45 47 75 13 91 51 9 53)) (G64 (99 27 15 45 19 7
67 63 25 61 19 13 63 81 57)) (G63 (43 69 83 37 35 63 45 11 17 69 97 67 79 9 83))
(G62 (95 75 7 21 15 73 79 73 57 79 23 21 79 33 65)) (G61 (9 17 77 23 59 77 15 73
67 21 33 13 79 89 49)) (G60 (95 69 93 99 83 51 31 85 57 37 85 15 33 49 41)) (G59
(93 95 51 23 25 19 99 27 43 43 77 53 19 41 77)) (G58 (79 23 23 27 49 77 21 59 35
27 3 75 15 81 85)))
Avg. turnaround time: 8
ThruPut: 0.32967032967033
```

```
The arrived list: ((G49 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G48 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G47 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G46 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G45 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G44 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G43 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G42 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G41 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G40 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G39 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G38 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G37 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G36 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G35 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G34 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G33 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G32 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G31 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G30 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G29 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G28 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G27 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G26 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G25 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G24 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G23 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G22 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G21 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G20 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G19 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G18 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G17 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G16 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G46 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G45 (49
75 11 47 37 1 21 31 19 5 69 61 75 39 29)) (G44 (49 59 11 85 97 67 61 29 83 29 9
63 5 57 31)) (G43 (3 67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G42 (45 21 23
71 73 39 25 99 91 11 85 3 19 97 31)) (G41 (37 11 75 53 15 79 47 37 25 99 71 5 93
73 45)) (G40 (95 97 13 1 39 23 25 53 71 57 25 3 79 7 21)) (G39 (11 77 37 43 1 21
69 25 79 37 95 41 83 43 5)) (G38 (75 47 59 53 71 37 73 21 1 91 35 87 49 99 77))
(G37 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G36 (99 39 71 17 61 61 61 99
1 89 91 27 95 41 45)) (G35 (35 69 23 61 15 21 59 13 87 67 9 9 67 71 99)) (G34 (
33 23 31 97 47 65 29 13 3 49 77 15 69 91 87)) (G33 (15 73 1 37 77 25 91 7 3 9 97
3 65 19 29)) (G32 (51 19 43 19 13 73 73 85 33 95 47 13 75 57 47)) (G31 (17 43 97
25 77 77 23 31 71 27 5 37 31 63 69)) (G30 (41 9 95 37 51 45 57 79 83 65 75 1 95
77 95)) (G29 (15 21 27 7 85 29 21 93 99 69 39 77 23 19 47)) (G28 (45 51 81 1 99
81 27 63 17 67 99 45 41 7 7)) (G27 (73 67 21 43 33 11 13 49 5 95 95 31 17 5 71))
(G26 (49 59 27 11 53 63 7 21 65 9 15 13 97 91 29)) (G25 (27 19 51 83 1 11 79 87
41 15 29 75 75 57 23)) (G24 (53 41 95 57 53 51 65 43 51 29 11 23 3 23 3)) (G23 (
29 59 3 79 15 93 95 13 95 19 61 3 1 45 93)) (G22 (9 93 63 53 43 75 29 37 87 11 3
79 69 83 1)) (G21 (1 39 35 31 93 27 23 47 41 27 31 21 7 89 19)) (G20 (47 69 11
13 95 49 81 53 19 33 67 73 55 7 85)) (G19 (3 83 93 17 7 97 7 63 57 73 9 7 95 83
45)) (G18 (81 65 11 51 93 51 89 9 81 25 83 83 53 5 37)) (G17 (97 41 37 67 9 85
37 95 51 55 29 97 89 73 27)) (G16 (47 79 95 75 49 75 19 87 41 81 35 15 17 61 83)
))
Avg. turnaround time: 8
ThruPut: 0.32967032967033
```

```
The arrived list: ((G49 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 99) (G48 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 96) (G47 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 93) (G46 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 90) (
G45 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 87) (G44 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 84) (G43 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 81)
(G42 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 78) (G41 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 75) (G40 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
72) (G39 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 69) (G38 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 66) (G37 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 63) (G36 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 60) (G35 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 57) (G34 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 54) (G33 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 51) (G32 (51 19 42 18
12 72 72 85 33 94 47 13 74 56 46) 48) (G31 (17 42 96 25 76 76 23 31 70 26 4 37
31 63 68) 45) (G30 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 42) (G29 (15 21
27 6 84 29 20 92 98 69 39 77 23 19 46) 39) (G28 (45 50 80 0 98 80 26 62 16 67 99
45 41 6 6) 36) (G27 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 33) (G26 (49 58
27 10 53 62 6 20 65 9 14 13 96 91 28) 30) (G25 (26 19 51 82 1 10 79 87 41 15 29
75 75 57 23) 27) (G24 (53 40 94 56 53 50 65 43 50 28 11 22 3 23 3) 24) (G23 (29
59 2 78 14 93 94 12 95 19 61 3 1 44 92) 21) (G22 (8 92 63 53 43 75 29 36 86 11 2
79 68 82 1) 18) (G21 (0 39 35 30 93 27 22 47 41 26 30 21 7 89 19) 15) (G20 (47
69 10 13 95 49 80 53 19 33 66 73 54 6 84) 12) (G19 (2 82 92 17 6 96 6 63 56 73 9
7 94 82 45) 9) (G18 (81 65 10 51 92 51 88 9 81 24 82 82 52 4 37) 6) (G17 (96 40
37 66 8 84 36 94 50 54 28 97 89 73 27) 3) (G16 (46 79 94 75 49 75 19 86 41 80 35
14 17 60 82) 0))
The solved list:  ((G46 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G45 (49
75 11 47 37 1 21 31 19 5 69 61 75 39 29)) (G44 (49 59 11 85 97 67 61 29 83 29 9
63 5 57 31)) (G43 (3 67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G42 (45 21 23
71 73 39 25 99 91 11 85 3 19 97 31)) (G41 (37 11 75 53 15 79 47 37 25 99 71 5 93
73 45)) (G40 (95 97 13 1 39 23 25 53 71 57 25 3 79 7 21)) (G39 (11 77 37 43 1 21
69 25 79 37 95 41 83 43 5)) (G38 (75 47 59 53 71 37 73 21 1 91 35 87 49 99 77))
(G37 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G36 (99 39 71 17 61 61 61 99
1 89 91 27 95 41 45)) (G35 (35 69 23 61 15 21 59 13 87 67 9 9 67 71 99)) (G34 (
33 23 31 97 47 65 29 13 3 49 77 15 69 91 87)) (G33 (15 73 1 37 77 25 91 7 3 9 97
3 65 19 29)) (G32 (51 19 43 19 13 73 73 85 33 95 47 13 75 57 47)) (G31 (17 43 97
25 77 77 23 31 71 27 5 37 31 63 69)) (G30 (41 9 95 37 51 45 57 79 83 65 75 1 95
77 95)) (G29 (15 21 27 7 85 29 21 93 99 69 39 77 23 19 47)) (G28 (45 51 81 1 99
81 27 63 17 67 99 45 41 7 7)) (G27 (73 67 21 43 33 11 13 49 5 95 95 31 17 5 71))
(G26 (49 59 27 11 53 63 7 21 65 9 15 13 97 91 29)) (G25 (27 19 51 83 1 11 79 87
41 15 29 75 75 57 23)) (G24 (53 41 95 57 53 51 65 43 51 29 11 23 3 23 3)) (G23 (
29 59 3 79 15 93 95 13 95 19 61 3 1 45 93)) (G22 (9 93 63 53 43 75 29 37 87 11 3
79 69 83 1)) (G21 (1 39 35 31 93 27 23 47 41 27 31 21 7 89 19)) (G20 (47 69 11
13 95 49 81 53 19 33 67 73 55 7 85)) (G19 (3 83 93 17 7 97 7 63 57 73 9 7 95 83
45)) (G18 (81 65 11 51 93 51 89 9 81 25 83 83 53 5 37)) (G17 (97 41 37 67 9 85
37 95 51 55 29 97 89 73 27)) (G16 (47 79 95 75 49 75 19 87 41 81 35 15 17 61 83)
))
Avg. turnaround time: 8
ThruPut: 0.32967032967033
```

3.1.3. ef-pip

```
The arrived list: ((G365 (59 66 32 70 97) 99) (G364 (95 74 3 77 14) 96) (G363 (
48 71 7 64 26) 93) (G362 (40 69 73 33 62) 90) (G361 (88 60 42 6 77) 87) (G360 (
54 40 5 10 22) 84) (G359 (75 10 68 12 35) 81) (G358 (20 95 58 23 0) 78) (G357 (
98 88 26 26 66) 75) (G356 (68 64 52 8 58) 72) (G355 (91 94 66 90 67) 69) (G354 (
36 0 80 74 15) 66) (G353 (22 22 42 72 65) 63) (G352 (8 78 88 35 22) 60) (G351 (
72 82 79 26 91) 57) (G350 (15 50 88 69 92) 54) (G349 (24 77 66 70 28) 51) (G348
(64 88 77 99 94) 48) (G347 (34 3 91 43 7) 45) (G346 (71 18 91 15 78) 42) (G345 (
56 49 46 82 97) 39) (G344 (17 67 38 47 91) 36) (G343 (97 68 30 64 41) 33) (G342
(40 90 0 64 76) 30) (G341 (73 80 88 62 30) 27) (G340 (85 89 91 64 2) 24) (G339 (
96 17 53 69 42) 21) (G338 (62 91 80 15 84) 18) (G337 (2 67 21 11 59) 15) (G336 (
91 7 11 65 20) 12) (G335 (46 49 79 51 43) 9) (G334 (72 63 54 74 30) 6) (G333 (2
4 24 58 73) 3) (G332 (66 6 1 80 29) 0))
The solved list:  ((G363 (49 71 7 65 27)) (G362 (41 69 73 33 63)) (G361 (89 61
43 7 77)) (G360 (55 41 5 11 23)) (G359 (75 11 69 13 35)) (G358 (21 95 59 23 1))
(G357 (99 89 27 27 67)) (G356 (69 65 53 9 59)) (G355 (91 95 67 91 67)) (G354 (37
1 81 75 15)) (G353 (23 23 43 73 65)) (G352 (9 79 89 35 23)) (G351 (73 83 79 27
91)) (G350 (15 51 89 69 93)) (G349 (25 77 67 71 29)) (G348 (65 89 77 99 95)) (
G347 (35 3 91 43 7)) (G346 (71 19 91 15 79)) (G345 (57 49 47 83 97)) (G344 (17
67 39 47 91)) (G343 (97 69 31 65 41)) (G342 (41 91 1 65 77)) (G341 (73 81 89 63
31)) (G340 (85 89 91 65 3)) (G339 (97 17 53 69 43)) (G338 (63 91 81 15 85)) (
G337 (3 67 21 11 59)) (G336 (91 7 11 65 21)) (G335 (47 49 79 51 43)) (G334 (73
63 55 75 31)) (G333 (3 5 25 59 73)) (G332 (67 7 1 81 29)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G49 (72 67 21 42 33) 99) (G48 (14 13 96 91 28) 96) (G47 (62
6 20 65 9) 93) (G46 (49 58 27 10 53) 90) (G45 (29 75 75 57 23) 87) (G44 (10 79
87 41 15) 84) (G43 (26 19 51 82 1) 81) (G42 (11 22 3 23 3) 78) (G41 (50 65 43 50
28) 75) (G40 (53 40 94 56 53) 72) (G39 (61 3 1 44 92) 69) (G38 (93 94 12 95 19)
66) (G37 (29 59 2 78 14) 63) (G36 (2 79 68 82 1) 60) (G35 (75 29 36 86 11) 57) (
G34 (8 92 63 53 43) 54) (G33 (30 21 7 89 19) 51) (G32 (27 22 47 41 26) 48) (G31
(0 39 35 30 93) 45) (G30 (66 73 54 6 84) 42) (G29 (49 80 53 19 33) 39) (G28 (47
69 10 13 95) 36) (G27 (9 7 94 82 45) 33) (G26 (96 6 63 56 73) 30) (G25 (2 82 92
17 6) 27) (G24 (82 82 52 4 37) 24) (G23 (51 88 9 81 24) 21) (G22 (81 65 10 51 92
) 18) (G21 (28 97 89 73 27) 15) (G20 (84 36 94 50 54) 12) (G19 (96 40 37 66 8) 9
) (G18 (35 14 17 60 82) 6) (G17 (75 19 86 41 80) 3) (G16 (46 79 94 75 49) 0))
The solved list:  ((G47 (63 7 21 65 9)) (G46 (49 59 27 11 53)) (G45 (29 75 75 57
23)) (G44 (11 79 87 41 15)) (G43 (27 19 51 83 1)) (G42 (11 23 3 23 3)) (G41 (51
65 43 51 29)) (G40 (53 41 95 57 53)) (G39 (61 3 1 45 93)) (G38 (93 95 13 95 19))
(G37 (29 59 3 79 15)) (G36 (3 79 69 83 1)) (G35 (75 29 37 87 11)) (G34 (9 93 63
53 43)) (G33 (31 21 7 89 19)) (G32 (27 23 47 41 27)) (G31 (1 39 35 31 93)) (G30
(67 73 55 7 85)) (G29 (49 81 53 19 33)) (G28 (47 69 11 13 95)) (G27 (9 7 95 83
45)) (G26 (97 7 63 57 73)) (G25 (3 83 93 17 7)) (G24 (83 83 53 5 37)) (G23 (51
89 9 81 25)) (G22 (81 65 11 51 93)) (G21 (29 97 89 73 27)) (G20 (85 37 95 51 55)
) (G19 (97 41 37 67 9)) (G18 (35 15 17 61 83)) (G17 (75 19 87 41 81)) (G16 (47
79 95 75 49)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G141 (33 4 82 60 29) 99) (G140 (37 26 4 57 76) 96) (G139 (17
25 27 35 7) 93) (G138 (90 45 29 60 91) 90) (G137 (63 36 55 95 23) 87) (G136 (50
80 7 34 36) 84) (G135 (95 16 4 48 59) 81) (G134 (57 13 8 43 12) 78) (G133 (87 44
88 1 23) 75) (G132 (86 91 83 96 58) 72) (G131 (77 32 90 59 97) 69) (G130 (2 40
48 83 33) 66) (G129 (87 3 6 33 57) 63) (G128 (72 77 48 0 73) 60) (G127 (3 66 22
31 53) 57) (G126 (5 13 98 52 56) 54) (G125 (6 37 41 42 48) 51) (G124 (94 12 97
70 95) 48) (G123 (53 40 67 86 36) 45) (G122 (55 38 15 82 57) 42) (G121 (23 29 16
1 84) 39) (G120 (86 55 57 34 43) 36) (G119 (87 68 9 52 1) 33) (G118 (2 68 90 7
54) 30) (G117 (80 46 52 39 74) 27) (G116 (14 29 41 24 12) 24) (G115 (91 20 48 58
85) 21) (G114 (74 51 15 89 81) 18) (G113 (7 31 61 84 14) 15) (G112 (81 90 24 4
16) 12) (G111 (94 98 14 8 62) 9) (G110 (93 6 46 52 31) 6) (G109 (95 52 75 17 65)
3) (G108 (34 65 70 52 60) 0))
The solved list:  ((G139 (17 25 27 35 7)) (G138 (91 45 29 61 91)) (G137 (63 37
55 95 23)) (G136 (51 81 7 35 37)) (G135 (95 17 5 49 59)) (G134 (57 13 9 43 13))
(G133 (87 45 89 1 23)) (G132 (87 91 83 97 59)) (G131 (77 33 91 59 97)) (G130 (3
41 49 83 33)) (G129 (87 3 7 33 57)) (G128 (73 77 49 1 73)) (G127 (3 67 23 31 53)
) (G126 (5 13 99 53 57)) (G125 (7 37 41 43 49)) (G124 (95 13 97 71 95)) (G123 (
53 41 67 87 37)) (G122 (55 39 15 83 57)) (G121 (23 29 17 1 85)) (G120 (87 55 57
35 43)) (G119 (87 69 9 53 1)) (G118 (3 69 91 7 55)) (G117 (81 47 53 39 75)) (
G116 (15 29 41 25 13)) (G115 (91 21 49 59 85)) (G114 (75 51 15 89 81)) (G113 (7
31 61 85 15)) (G112 (81 91 25 5 17)) (G111 (95 99 15 9 63)) (G110 (93 7 47 53 31
)) (G109 (95 53 75 17 65)) (G108 (35 65 71 53 61)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G49 (72 67 21 42 33) 99) (G48 (14 13 96 91 28) 96) (G47 (62
6 20 65 9) 93) (G46 (49 58 27 10 53) 90) (G45 (29 75 75 57 23) 87) (G44 (10 79
87 41 15) 84) (G43 (26 19 51 82 1) 81) (G42 (11 22 3 23 3) 78) (G41 (50 65 43 50
28) 75) (G40 (53 40 94 56 53) 72) (G39 (61 3 1 44 92) 69) (G38 (93 94 12 95 19)
66) (G37 (29 59 2 78 14) 63) (G36 (2 79 68 82 1) 60) (G35 (75 29 36 86 11) 57) (
G34 (8 92 63 53 43) 54) (G33 (30 21 7 89 19) 51) (G32 (27 22 47 41 26) 48) (G31
(0 39 35 30 93) 45) (G30 (66 73 54 6 84) 42) (G29 (49 80 53 19 33) 39) (G28 (47
69 10 13 95) 36) (G27 (9 7 94 82 45) 33) (G26 (96 6 63 56 73) 30) (G25 (2 82 92
17 6) 27) (G24 (82 82 52 4 37) 24) (G23 (51 88 9 81 24) 21) (G22 (81 65 10 51 92
) 18) (G21 (28 97 89 73 27) 15) (G20 (84 36 94 50 54) 12) (G19 (96 40 37 66 8) 9
) (G18 (35 14 17 60 82) 6) (G17 (75 19 86 41 80) 3) (G16 (46 79 94 75 49) 0))
The solved list:  ((G47 (63 7 21 65 9)) (G46 (49 59 27 11 53)) (G45 (29 75 75 57
23)) (G44 (11 79 87 41 15)) (G43 (27 19 51 83 1)) (G42 (11 23 3 23 3)) (G41 (51
65 43 51 29)) (G40 (53 41 95 57 53)) (G39 (61 3 1 45 93)) (G38 (93 95 13 95 19))
(G37 (29 59 3 79 15)) (G36 (3 79 69 83 1)) (G35 (75 29 37 87 11)) (G34 (9 93 63
53 43)) (G33 (31 21 7 89 19)) (G32 (27 23 47 41 27)) (G31 (1 39 35 31 93)) (G30
(67 73 55 7 85)) (G29 (49 81 53 19 33)) (G28 (47 69 11 13 95)) (G27 (9 7 95 83
45)) (G26 (97 7 63 57 73)) (G25 (3 83 93 17 7)) (G24 (83 83 53 5 37)) (G23 (51
89 9 81 25)) (G22 (81 65 11 51 93)) (G21 (29 97 89 73 27)) (G20 (85 37 95 51 55)
) (G19 (97 41 37 67 9)) (G18 (35 15 17 61 83)) (G17 (75 19 87 41 81)) (G16 (47
79 95 75 49)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G49 (72 67 21 42 33) 99) (G48 (14 13 96 91 28) 96) (G47 (62
6 20 65 9) 93) (G46 (49 58 27 10 53) 90) (G45 (29 75 75 57 23) 87) (G44 (10 79
87 41 15) 84) (G43 (26 19 51 82 1) 81) (G42 (11 22 3 23 3) 78) (G41 (50 65 43 50
28) 75) (G40 (53 40 94 56 53) 72) (G39 (61 3 1 44 92) 69) (G38 (93 94 12 95 19)
66) (G37 (29 59 2 78 14) 63) (G36 (2 79 68 82 1) 60) (G35 (75 29 36 86 11) 57) (
G34 (8 92 63 53 43) 54) (G33 (30 21 7 89 19) 51) (G32 (27 22 47 41 26) 48) (G31
(0 39 35 30 93) 45) (G30 (66 73 54 6 84) 42) (G29 (49 80 53 19 33) 39) (G28 (47
69 10 13 95) 36) (G27 (9 7 94 82 45) 33) (G26 (96 6 63 56 73) 30) (G25 (2 82 92
17 6) 27) (G24 (82 82 52 4 37) 24) (G23 (51 88 9 81 24) 21) (G22 (81 65 10 51 92
) 18) (G21 (28 97 89 73 27) 15) (G20 (84 36 94 50 54) 12) (G19 (96 40 37 66 8) 9
) (G18 (35 14 17 60 82) 6) (G17 (75 19 86 41 80) 3) (G16 (46 79 94 75 49) 0))
The solved list:  ((G47 (63 7 21 65 9)) (G46 (49 59 27 11 53)) (G45 (29 75 75 57
23)) (G44 (11 79 87 41 15)) (G43 (27 19 51 83 1)) (G42 (11 23 3 23 3)) (G41 (51
65 43 51 29)) (G40 (53 41 95 57 53)) (G39 (61 3 1 45 93)) (G38 (93 95 13 95 19))
(G37 (29 59 3 79 15)) (G36 (3 79 69 83 1)) (G35 (75 29 37 87 11)) (G34 (9 93 63
53 43)) (G33 (31 21 7 89 19)) (G32 (27 23 47 41 27)) (G31 (1 39 35 31 93)) (G30
(67 73 55 7 85)) (G29 (49 81 53 19 33)) (G28 (47 69 11 13 95)) (G27 (9 7 95 83
45)) (G26 (97 7 63 57 73)) (G25 (3 83 93 17 7)) (G24 (83 83 53 5 37)) (G23 (51
89 9 81 25)) (G22 (81 65 11 51 93)) (G21 (29 97 89 73 27)) (G20 (85 37 95 51 55)
) (G19 (97 41 37 67 9)) (G18 (35 15 17 61 83)) (G17 (75 19 87 41 81)) (G16 (47
79 95 75 49)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

3.1.4. ef-dc

```
The arrived list: ((G265 (29 52 83 20 25 62 70 19 64 99 37 6 96 91 81) 99) (G264
(45 13 55 31 56 35 83 58 23 0 24 23 26 18 48) 96) (G263 (61 9 90 35 28 22 98 71
39 64 27 49 65 13 94) 93) (G262 (1 80 20 17 70 84 16 61 87 50 87 4 5 62 51) 90)
(G261 (37 83 33 37 55 37 72 18 94 87 20 72 53 58 54) 87) (G260 (41 32 49 93 50
40 50 28 50 66 19 82 65 24 96) 84) (G259 (79 69 12 6 7 35 48 77 69 15 76 14 43
79 73) 81) (G258 (36 34 16 97 57 22 37 56 32 2 15 63 97 10 35) 78) (G257 (86 55
14 81 34 97 14 50 45 61 61 29 66 68 3) 75) (G256 (5 13 68 37 99 83 30 87 46 52
50 50 86 95 47) 72) (G255 (83 2 60 52 71 34 32 96 34 61 46 50 23 50 83) 69) (
G254 (39 74 58 50 94 98 93 77 10 79 69 65 74 81 17) 66) (G253 (57 86 30 62 73 85
12 1 42 27 42 61 63 80 27) 63) (G252 (71 15 91 40 58 49 92 57 70 58 48 96 99 65
61) 60) (G251 (49 82 22 33 15 31 11 74 13 59 91 42 18 78 30) 57) (G250 (48 63 14
9 18 25 5 80 88 53 63 97 8 64 36) 54) (G249 (33 24 1 43 72 7 14 90 63 62 97 87
33 77 68) 51) (G248 (46 39 73 67 5 50 87 25 57 49 50 93 9 87 82) 48) (G247 (11
77 17 30 79 73 19 61 55 40 57 73 20 21 70) 45) (G246 (42 42 66 40 90 25 25 31 97
41 28 4 13 38 29) 42) (G245 (14 15 6 32 92 31 59 52 3 83 5 97 80 81 84) 39) (
G244 (69 92 98 33 84 74 33 40 43 44 86 34 37 53 72) 36) (G243 (96 50 71 29 54 97
61 21 29 53 62 90 9 3 97) 33) (G242 (38 67 70 52 42 51 23 86 11 54 17 23 98 5 5)
30) (G241 (12 25 75 36 79 80 32 31 18 6 42 30 98 95 45) 27) (G240 (69 34 30 61
28 42 27 19 22 38 6 71 89 9 86) 24) (G239 (15 76 25 42 32 46 56 94 95 36 96 14
81 81 34) 21) (G238 (50 92 17 52 79 82 60 17 69 16 31 6 74 70 62) 18) (G237 (76
13 16 55 75 23 38 54 12 43 71 22 41 97 23) 15) (G236 (43 94 4 24 77 95 28 56 18
97 39 4 51 23 67) 12) (G235 (64 20 21 82 70 91 22 91 0 68 56 5 15 11 21) 9) (
G234 (70 66 25 2 1 84 12 68 41 70 52 10 72 35 69) 6) (G233 (35 32 50 66 15 29 13
8 51 98 0 65 90 80 12) 3) (G232 (28 42 40 56 16 58 33 34 80 40 70 39 69 21 80) 0
))
The solved list:  ((G263 (61 9 91 35 29 23 99 71 39 65 27 49 65 13 95)) (G262 (1
81 21 17 71 85 17 61 87 51 87 5 5 63 51)) (G261 (37 83 33 37 55 37 73 19 95 87
21 73 53 59 55)) (G260 (41 33 49 93 51 41 51 29 51 67 19 83 65 25 97)) (G259 (79
69 13 7 7 35 49 77 69 15 77 15 43 79 73)) (G258 (37 35 17 97 57 23 37 57 33 3 15
63 97 11 35)) (G257 (87 55 15 81 35 97 15 51 45 61 61 29 67 69 3)) (G256 (5 13
69 37 99 83 31 87 47 53 51 51 87 95 47)) (G255 (83 3 61 53 71 35 33 97 35 61 47
51 23 51 83)) (G254 (39 75 59 51 95 99 93 77 11 79 69 65 75 81 17)) (G253 (57 87
31 63 73 85 13 1 43 27 43 61 63 81 27)) (G252 (71 15 91 41 59 49 93 57 71 59 49
97 99 65 61)) (G251 (49 83 23 33 15 31 11 75 13 59 91 43 19 79 31)) (G250 (49 63
15 9 19 25 5 81 89 53 63 97 9 65 37)) (G249 (33 25 1 43 73 7 15 91 63 63 97 87
33 77 69)) (G248 (47 39 73 67 5 51 87 25 57 49 51 93 9 87 83)) (G247 (11 77 17
31 79 73 19 61 55 41 57 73 21 21 71)) (G246 (43 43 67 41 91 25 25 31 97 41 29 5
13 39 29)) (G245 (15 15 7 33 93 31 59 53 3 83 5 97 81 81 85)) (G244 (69 93 99 33
85 75 33 41 43 45 87 35 37 53 73)) (G243 (97 51 71 29 55 97 61 21 29 53 63 91 9
3 97)) (G242 (39 67 71 53 43 51 23 87 11 55 17 23 99 5 5)) (G241 (13 25 75 37 79
81 33 31 19 7 43 31 99 95 45)) (G240 (69 35 31 61 29 43 27 19 23 39 7 71 89 9 87
)) (G239 (15 77 25 43 33 47 57 95 95 37 97 15 81 81 35)) (G238 (51 93 17 53 79
83 61 17 69 17 31 7 75 71 63)) (G237 (77 13 17 55 75 23 39 55 13 43 71 23 41 97
23)) (G236 (43 95 5 25 77 95 29 57 19 97 39 5 51 23 67)) (G235 (65 21 21 83 71
91 23 91 1 69 57 5 15 11 21)) (G234 (71 67 25 3 1 85 13 69 41 71 53 11 73 35 69)
) (G233 (35 33 51 67 15 29 13 9 51 99 1 65 91 81 13)) (G232 (29 43 41 57 17 59
33 35 81 41 71 39 69 21 81)))
Avg. turnaround time: 7
ThruPut: 0.333333333333333
```

```
The arrived list: ((G115 (86 44 6 71 40 24 41 8 86 29 5 34 47 8 92) 99) (G114 (
49 57 9 68 48 97 6 2 47 61 60 79 10 20 5) 96) (G113 (53 41 32 86 40 9 91 8 81 30
4 74 52 60 35) 93) (G112 (89 66 74 75 53 86 96 77 22 20 79 26 36 66 60) 90) (
G111 (53 22 29 63 89 40 63 98 86 25 51 22 12 65 21) 87) (G110 (0 19 94 22 74 32
13 69 39 30 6 45 91 87 59) 84) (G109 (4 30 63 20 55 68 2 22 55 3 41 41 67 53 84)
81) (G108 (15 16 78 28 21 40 49 86 13 63 17 25 47 84 76) 78) (G107 (75 5 38 40
27 81 89 11 46 15 93 3 52 5 80) 75) (G106 (56 91 47 83 10 41 69 49 74 57 66 45
63 53 39) 72) (G105 (60 19 15 21 47 39 84 76 40 49 26 49 15 23 72) 69) (G104 (10
68 13 45 71 24 26 78 16 19 97 99 90 19 9) 66) (G103 (3 32 27 33 11 39 95 29 34
42 77 63 36 92 86) 63) (G102 (50 20 72 40 77 31 24 40 2 49 86 9 92 80 19) 60) (
G101 (54 66 85 75 64 15 83 35 40 64 17 82 53 85 89) 57) (G100 (89 11 7 56 19 90
56 31 32 21 94 91 49 75 41) 54) (G99 (34 88 41 66 22 17 1 85 27 43 26 73 33 78
76) 51) (G98 (25 69 12 94 83 23 32 98 79 7 70 67 92 13 98) 48) (G97 (21 67 23 40
93 55 58 41 7 52 48 13 22 8 84) 45) (G96 (0 40 96 74 85 15 23 8 72 43 3 51 85 77
5) 42) (G95 (47 77 84 52 65 30 35 48 8 6 81 61 14 10 44) 39) (G94 (86 66 75 76
23 10 8 78 64 45 95 64 50 55 7) 36) (G93 (31 4 92 40 83 74 36 95 18 85 96 78 45
33 12) 33) (G92 (52 96 73 70 52 95 60 77 87 71 68 44 73 23 76) 30) (G91 (97 51
73 3 9 79 16 98 62 96 76 64 57 89 55) 27) (G90 (88 18 95 75 39 9 98 13 82 35 56
53 38 90 73) 24) (G89 (67 64 22 72 28 40 74 45 46 75 12 91 51 9 53) 21) (G88 (99
26 15 44 19 7 67 62 24 60 18 12 63 80 56) 18) (G87 (43 69 83 36 34 63 44 10 17
68 96 67 79 9 83) 15) (G86 (94 75 7 21 14 72 78 73 56 78 23 20 79 32 64) 12) (
G85 (9 16 76 23 58 77 15 73 67 21 33 13 79 89 48) 9) (G84 (94 69 92 99 82 50 30
85 57 37 85 14 33 49 41) 6) (G83 (93 95 50 23 24 19 98 27 43 42 76 52 18 40 76)
3) (G82 (78 23 23 26 49 76 21 59 34 26 3 74 15 80 84) 0))
The solved list:  ((G113 (53 41 33 87 41 9 91 9 81 31 5 75 53 61 35)) (G112 (89
67 75 75 53 87 97 77 23 21 79 27 37 67 61)) (G111 (53 23 29 63 89 41 63 99 87 25
51 23 13 65 21)) (G110 (1 19 95 23 75 33 13 69 39 31 7 45 91 87 59)) (G109 (5 31
63 21 55 69 3 23 55 3 41 41 67 53 85)) (G108 (15 17 79 29 21 41 49 87 13 63 17
25 47 85 77)) (G107 (75 5 39 41 27 81 89 11 47 15 93 3 53 5 81)) (G106 (57 91 47
83 11 41 69 49 75 57 67 45 63 53 39)) (G105 (61 19 15 21 47 39 85 77 41 49 27 49
15 23 73)) (G104 (11 69 13 45 71 25 27 79 17 19 97 99 91 19 9)) (G103 (3 33 27
33 11 39 95 29 35 43 77 63 37 93 87)) (G102 (51 21 73 41 77 31 25 41 3 49 87 9
93 81 19)) (G101 (55 67 85 75 65 15 83 35 41 65 17 83 53 85 89)) (G100 (89 11 7
57 19 91 57 31 33 21 95 91 49 75 41)) (G99 (35 89 41 67 23 17 1 85 27 43 27 73
33 79 77)) (G98 (25 69 13 95 83 23 33 99 79 7 71 67 93 13 99)) (G97 (21 67 23 41
93 55 59 41 7 53 49 13 23 9 85)) (G96 (1 41 97 75 85 15 23 9 73 43 3 51 85 77 5)
) (G95 (47 77 85 53 65 31 35 49 9 7 81 61 15 11 45)) (G94 (87 67 75 77 23 11 9
79 65 45 95 65 51 55 7)) (G93 (31 5 93 41 83 75 37 95 19 85 97 79 45 33 13)) (
G92 (53 97 73 71 53 95 61 77 87 71 69 45 73 23 77)) (G91 (97 51 73 3 9 79 17 99
63 97 77 65 57 89 55)) (G90 (89 19 95 75 39 9 99 13 83 35 57 53 39 91 73)) (G89
(67 65 23 73 29 41 75 45 47 75 13 91 51 9 53)) (G88 (99 27 15 45 19 7 67 63 25
61 19 13 63 81 57)) (G87 (43 69 83 37 35 63 45 11 17 69 97 67 79 9 83)) (G86 (95
75 7 21 15 73 79 73 57 79 23 21 79 33 65)) (G85 (9 17 77 23 59 77 15 73 67 21 33
13 79 89 49)) (G84 (95 69 93 99 83 51 31 85 57 37 85 15 33 49 41)) (G83 (93 95
51 23 25 19 99 27 43 43 77 53 19 41 77)) (G82 (79 23 23 27 49 77 21 59 35 27 3
75 15 81 85)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G408 (26 32 67 76 58 22 6 20 32 50 82 39 86 43 70) 99) (G407
(87 96 98 12 3 1 59 23 40 21 99 4 18 59 92) 96) (G406 (28 35 88 18 12 53 50 14
15 57 79 93 16 2 25) 93) (G405 (57 23 71 40 87 33 37 74 81 87 70 8 8 32 40) 90)
(G404 (8 50 11 52 59 84 26 32 16 74 97 10 6 41 43) 87) (G403 (90 85 20 23 35 67
80 64 62 66 97 54 16 78 83) 84) (G402 (7 85 54 12 4 92 10 67 14 85 64 21 98 98
24) 81) (G401 (53 47 67 7 31 63 68 59 7 3 93 17 32 59 55) 78) (G400 (8 11 70 4
76 35 52 63 26 51 76 47 85 48 33) 75) (G399 (81 79 77 1 48 23 62 26 26 84 0 39
83 64 50) 72) (G398 (35 29 32 22 99 73 66 98 33 22 99 91 79 34 25) 69) (G397 (24
59 31 86 44 75 21 63 44 9 23 22 44 6 98) 66) (G396 (25 26 45 52 65 9 67 55 61 14
34 78 17 91 33) 63) (G395 (57 86 73 29 39 2 91 66 0 96 76 90 2 57 80) 60) (G394
(0 99 3 69 63 14 52 80 29 84 20 2 23 55 93) 57) (G393 (30 96 5 72 39 7 65 62 23
40 34 68 39 66 91) 54) (G392 (42 70 73 99 47 50 66 73 8 52 38 59 1 82 43) 51) (
G391 (41 4 47 53 88 25 87 57 61 82 28 69 6 85 44) 48) (G390 (85 4 55 88 35 28 22
9 97 1 24 90 91 98 59) 45) (G389 (42 1 1 48 56 45 21 32 35 60 98 19 3 17 50) 42)
(G388 (1 38 67 41 78 87 17 23 28 57 37 15 53 76 9) 39) (G387 (58 46 35 50 4 27
39 24 2 14 85 59 49 22 64) 36) (G386 (69 43 71 81 4 35 19 32 14 19 36 12 87 88
69) 33) (G385 (60 25 45 54 98 58 28 39 79 0 94 3 57 87 78) 30) (G384 (6 48 10 41
5 72 48 27 31 40 97 56 76 60 24) 27) (G383 (49 22 51 23 62 78 89 15 65 26 32 11
10 91 0) 24) (G382 (94 30 13 89 2 58 44 78 50 87 53 46 40 8 75) 21) (G381 (50 46
85 39 21 84 47 54 64 23 2 89 25 19 37) 18) (G380 (1 14 64 46 5 12 98 7 75 15 17
14 77 73 21) 15) (G379 (9 16 41 53 8 44 94 97 8 57 57 87 33 95 70) 12) (G378 (8
31 81 53 93 55 56 42 41 59 42 57 89 32 87) 9) (G377 (5 0 61 83 90 98 41 17 4 28
97 58 13 99 25) 6) (G376 (5 6 84 81 74 15 85 78 30 39 83 57 14 98 6) 3) (G375 (
59 66 32 70 97 11 33 23 5 8 27 95 77 87 60) 0))
The solved list:  ((G406 (29 35 89 19 13 53 51 15 15 57 79 93 17 3 25)) (G405 (
57 23 71 41 87 33 37 75 81 87 71 9 9 33 41)) (G404 (9 51 11 53 59 85 27 33 17 75
97 11 7 41 43)) (G403 (91 85 21 23 35 67 81 65 63 67 97 55 17 79 83)) (G402 (7
85 55 13 5 93 11 67 15 85 65 21 99 99 25)) (G401 (53 47 67 7 31 63 69 59 7 3 93
17 33 59 55)) (G400 (9 11 71 5 77 35 53 63 27 51 77 47 85 49 33)) (G399 (81 79
77 1 49 23 63 27 27 85 1 39 83 65 51)) (G398 (35 29 33 23 99 73 67 99 33 23 99
91 79 35 25)) (G397 (25 59 31 87 45 75 21 63 45 9 23 23 45 7 99)) (G396 (25 27
45 53 65 9 67 55 61 15 35 79 17 91 33)) (G395 (57 87 73 29 39 3 91 67 1 97 77 91
3 57 81)) (G394 (1 99 3 69 63 15 53 81 29 85 21 3 23 55 93)) (G393 (31 97 5 73
39 7 65 63 23 41 35 69 39 67 91)) (G392 (43 71 73 99 47 51 67 73 9 53 39 59 1 83
43)) (G391 (41 5 47 53 89 25 87 57 61 83 29 69 7 85 45)) (G390 (85 5 55 89 35 29
23 9 97 1 25 91 91 99 59)) (G389 (43 1 1 49 57 45 21 33 35 61 99 19 3 17 51)) (
G388 (1 39 67 41 79 87 17 23 29 57 37 15 53 77 9)) (G387 (59 47 35 51 5 27 39 25
3 15 85 59 49 23 65)) (G386 (69 43 71 81 5 35 19 33 15 19 37 13 87 89 69)) (G385
(61 25 45 55 99 59 29 39 79 1 95 3 57 87 79)) (G384 (7 49 11 41 5 73 49 27 31 41
97 57 77 61 25)) (G383 (49 23 51 23 63 79 89 15 65 27 33 11 11 91 1)) (G382 (95
31 13 89 3 59 45 79 51 87 53 47 41 9 75)) (G381 (51 47 85 39 21 85 47 55 65 23 3
89 25 19 37)) (G380 (1 15 65 47 5 13 99 7 75 15 17 15 77 73 21)) (G379 (9 17 41
53 9 45 95 97 9 57 57 87 33 95 71)) (G378 (9 31 81 53 93 55 57 43 41 59 43 57 89
33 87)) (G377 (5 1 61 83 91 99 41 17 5 29 97 59 13 99 25)) (G376 (5 7 85 81 75
15 85 79 31 39 83 57 15 99 7)) (G375 (59 67 33 71 97 11 33 23 5 9 27 95 77 87 61
)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G90 (88 18 95 75 39 9 98 13 82 35 56 53 38 90 73) 99) (G89 (
67 64 22 72 28 40 74 45 46 75 12 91 51 9 53) 96) (G88 (99 26 15 44 19 7 67 62 24
60 18 12 63 80 56) 93) (G87 (43 69 83 36 34 63 44 10 17 68 96 67 79 9 83) 90) (
G86 (94 75 7 21 14 72 78 73 56 78 23 20 79 32 64) 87) (G85 (9 16 76 23 58 77 15
73 67 21 33 13 79 89 48) 84) (G84 (94 69 92 99 82 50 30 85 57 37 85 14 33 49 41)
81) (G83 (93 95 50 23 24 19 98 27 43 42 76 52 18 40 76) 78) (G82 (78 23 23 26 49
76 21 59 34 26 3 74 15 80 84) 75) (G81 (46 54 29 94 19 33 33 27 47 84 31 69 9 83
63) 72) (G80 (68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 69) (G79 (39 91 12 78
48 27 89 53 10 82 10 35 43 97 22) 66) (G78 (18 15 18 24 18 88 50 78 88 32 17 12
92 90 56) 63) (G77 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 60) (G76 (49 59
11 84 96 67 61 28 82 28 8 63 4 56 30) 57) (G75 (3 66 14 93 0 25 85 26 46 60 69
20 93 72 20) 54) (G74 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 51) (G73 (36
10 75 53 14 79 46 36 24 98 70 4 93 73 45) 48) (G72 (94 96 13 1 39 23 25 52 71 57
25 2 78 7 21) 45) (G71 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 42) (G70 (75
47 59 53 70 36 73 21 1 90 34 86 49 98 76) 39) (G69 (78 15 63 12 4 1 47 20 14 54
68 37 62 80 52) 36) (G68 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 33) (G67
(34 69 22 61 15 21 58 13 87 66 8 9 67 70 98) 30) (G66 (32 22 30 96 47 64 28 12 2
49 77 15 68 91 86) 27) (G65 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 24) (G64 (
51 19 42 18 12 72 72 85 33 94 47 13 74 56 46) 21) (G63 (17 42 96 25 76 76 23 31
70 26 4 37 31 63 68) 18) (G62 (41 8 95 37 50 45 57 79 83 64 74 0 94 76 95) 15) (
G61 (15 21 27 6 84 29 20 92 98 69 39 77 23 19 46) 12) (G60 (45 50 80 0 98 80 26
62 16 67 99 45 41 6 6) 9) (G59 (72 67 21 42 33 11 13 49 4 94 95 31 17 4 70) 6) (
G58 (49 58 27 10 53 62 6 20 65 9 14 13 96 91 28) 3) (G57 (26 19 51 82 1 10 79 87
41 15 29 75 75 57 23) 0))
The solved list:  ((G88 (99 27 15 45 19 7 67 63 25 61 19 13 63 81 57)) (G87 (43
69 83 37 35 63 45 11 17 69 97 67 79 9 83)) (G86 (95 75 7 21 15 73 79 73 57 79 23
21 79 33 65)) (G85 (9 17 77 23 59 77 15 73 67 21 33 13 79 89 49)) (G84 (95 69 93
99 83 51 31 85 57 37 85 15 33 49 41)) (G83 (93 95 51 23 25 19 99 27 43 43 77 53
19 41 77)) (G82 (79 23 23 27 49 77 21 59 35 27 3 75 15 81 85)) (G81 (47 55 29 95
19 33 33 27 47 85 31 69 9 83 63)) (G80 (69 5 73 35 87 69 67 65 15 85 75 3 75 55
57)) (G79 (39 91 13 79 49 27 89 53 11 83 11 35 43 97 23)) (G78 (19 15 19 25 19
89 51 79 89 33 17 13 93 91 57)) (G77 (49 75 11 47 37 1 21 31 19 5 69 61 75 39 29
)) (G76 (49 59 11 85 97 67 61 29 83 29 9 63 5 57 31)) (G75 (3 67 15 93 1 25 85
27 47 61 69 21 93 73 21)) (G74 (45 21 23 71 73 39 25 99 91 11 85 3 19 97 31)) (
G73 (37 11 75 53 15 79 47 37 25 99 71 5 93 73 45)) (G72 (95 97 13 1 39 23 25 53
71 57 25 3 79 7 21)) (G71 (11 77 37 43 1 21 69 25 79 37 95 41 83 43 5)) (G70 (75
47 59 53 71 37 73 21 1 91 35 87 49 99 77)) (G69 (79 15 63 13 5 1 47 21 15 55 69
37 63 81 53)) (G68 (99 39 71 17 61 61 61 99 1 89 91 27 95 41 45)) (G67 (35 69 23
61 15 21 59 13 87 67 9 9 67 71 99)) (G66 (33 23 31 97 47 65 29 13 3 49 77 15 69
91 87)) (G65 (15 73 1 37 77 25 91 7 3 9 97 3 65 19 29)) (G64 (51 19 43 19 13 73
73 85 33 95 47 13 75 57 47)) (G63 (17 43 97 25 77 77 23 31 71 27 5 37 31 63 69))
(G62 (41 9 95 37 51 45 57 79 83 65 75 1 95 77 95)) (G61 (15 21 27 7 85 29 21 93
99 69 39 77 23 19 47)) (G60 (45 51 81 1 99 81 27 63 17 67 99 45 41 7 7)) (G59 (
73 67 21 43 33 11 13 49 5 95 95 31 17 5 71)) (G58 (49 59 27 11 53 63 7 21 65 9
15 13 97 91 29)) (G57 (27 19 51 83 1 11 79 87 41 15 29 75 75 57 23)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

```
The arrived list: ((G97 (21 67 23 40 93 55 58 41 7 52 48 13 22 8 84) 99) (G96 (0
40 96 74 85 15 23 8 72 43 3 51 85 77 5) 96) (G95 (47 77 84 52 65 30 35 48 8 6 81
61 14 10 44) 93) (G94 (86 66 75 76 23 10 8 78 64 45 95 64 50 55 7) 90) (G93 (31
4 92 40 83 74 36 95 18 85 96 78 45 33 12) 87) (G92 (52 96 73 70 52 95 60 77 87
71 68 44 73 23 76) 84) (G91 (97 51 73 3 9 79 16 98 62 96 76 64 57 89 55) 81) (
G90 (88 18 95 75 39 9 98 13 82 35 56 53 38 90 73) 78) (G89 (67 64 22 72 28 40 74
45 46 75 12 91 51 9 53) 75) (G88 (99 26 15 44 19 7 67 62 24 60 18 12 63 80 56)
72) (G87 (43 69 83 36 34 63 44 10 17 68 96 67 79 9 83) 69) (G86 (94 75 7 21 14
72 78 73 56 78 23 20 79 32 64) 66) (G85 (9 16 76 23 58 77 15 73 67 21 33 13 79
89 48) 63) (G84 (94 69 92 99 82 50 30 85 57 37 85 14 33 49 41) 60) (G83 (93 95
50 23 24 19 98 27 43 42 76 52 18 40 76) 57) (G82 (78 23 23 26 49 76 21 59 34 26
3 74 15 80 84) 54) (G81 (46 54 29 94 19 33 33 27 47 84 31 69 9 83 63) 51) (G80 (
68 4 72 35 87 68 66 65 15 84 75 3 75 54 56) 48) (G79 (39 91 12 78 48 27 89 53 10
82 10 35 43 97 22) 45) (G78 (18 15 18 24 18 88 50 78 88 32 17 12 92 90 56) 42) (
G77 (48 74 10 46 36 0 20 30 19 5 69 60 74 39 28) 39) (G76 (49 59 11 84 96 67 61
28 82 28 8 63 4 56 30) 36) (G75 (3 66 14 93 0 25 85 26 46 60 69 20 93 72 20) 33)
(G74 (45 21 22 70 73 38 24 99 91 10 84 3 18 97 31) 30) (G73 (36 10 75 53 14 79
46 36 24 98 70 4 93 73 45) 27) (G72 (94 96 13 1 39 23 25 52 71 57 25 2 78 7 21)
24) (G71 (11 77 37 42 0 20 68 24 78 37 95 41 82 43 4) 21) (G70 (75 47 59 53 70
36 73 21 1 90 34 86 49 98 76) 18) (G69 (78 15 63 12 4 1 47 20 14 54 68 37 62 80
52) 15) (G68 (98 39 70 16 61 60 60 99 0 89 91 26 94 41 44) 12) (G67 (34 69 22 61
15 21 58 13 87 66 8 9 67 70 98) 9) (G66 (32 22 30 96 47 64 28 12 2 49 77 15 68
91 86) 6) (G65 (15 72 0 37 76 24 91 6 3 9 96 3 64 18 28) 3) (G64 (51 19 42 18 12
72 72 85 33 94 47 13 74 56 46) 0))
The solved list:  ((G95 (47 77 85 53 65 31 35 49 9 7 81 61 15 11 45)) (G94 (87
67 75 77 23 11 9 79 65 45 95 65 51 55 7)) (G93 (31 5 93 41 83 75 37 95 19 85 97
79 45 33 13)) (G92 (53 97 73 71 53 95 61 77 87 71 69 45 73 23 77)) (G91 (97 51
73 3 9 79 17 99 63 97 77 65 57 89 55)) (G90 (89 19 95 75 39 9 99 13 83 35 57 53
39 91 73)) (G89 (67 65 23 73 29 41 75 45 47 75 13 91 51 9 53)) (G88 (99 27 15 45
19 7 67 63 25 61 19 13 63 81 57)) (G87 (43 69 83 37 35 63 45 11 17 69 97 67 79 9
83)) (G86 (95 75 7 21 15 73 79 73 57 79 23 21 79 33 65)) (G85 (9 17 77 23 59 77
15 73 67 21 33 13 79 89 49)) (G84 (95 69 93 99 83 51 31 85 57 37 85 15 33 49 41)
) (G83 (93 95 51 23 25 19 99 27 43 43 77 53 19 41 77)) (G82 (79 23 23 27 49 77
21 59 35 27 3 75 15 81 85)) (G81 (47 55 29 95 19 33 33 27 47 85 31 69 9 83 63))
(G80 (69 5 73 35 87 69 67 65 15 85 75 3 75 55 57)) (G79 (39 91 13 79 49 27 89 53
11 83 11 35 43 97 23)) (G78 (19 15 19 25 19 89 51 79 89 33 17 13 93 91 57)) (G77
(49 75 11 47 37 1 21 31 19 5 69 61 75 39 29)) (G76 (49 59 11 85 97 67 61 29 83
29 9 63 5 57 31)) (G75 (3 67 15 93 1 25 85 27 47 61 69 21 93 73 21)) (G74 (45 21
23 71 73 39 25 99 91 11 85 3 19 97 31)) (G73 (37 11 75 53 15 79 47 37 25 99 71 5
93 73 45)) (G72 (95 97 13 1 39 23 25 53 71 57 25 3 79 7 21)) (G71 (11 77 37 43 1
21 69 25 79 37 95 41 83 43 5)) (G70 (75 47 59 53 71 37 73 21 1 91 35 87 49 99 77
)) (G69 (79 15 63 13 5 1 47 21 15 55 69 37 63 81 53)) (G68 (99 39 71 17 61 61 61
99 1 89 91 27 95 41 45)) (G67 (35 69 23 61 15 21 59 13 87 67 9 9 67 71 99)) (G66
(33 23 31 97 47 65 29 13 3 49 77 15 69 91 87)) (G65 (15 73 1 37 77 25 91 7 3 9
97 3 65 19 29)) (G64 (51 19 43 19 13 73 73 85 33 95 47 13 75 57 47)))
Avg. turnaround time: 6
ThruPut: 0.333333333333333
```

4. model program files

4.1. pl.m

```
(load "mbase/func.m")
  (make-pair atomic-models 'pl)
  (send pl def-state
          '(
            job-id         ;name of the processed job
            processing-time  ; processing time of this processor
           )
  )
  (send pl set-s
          (make-state 'sigma     'inf
                      'phase     'passive
                      'job-id    '()
                      'processing-time rnd-seed
          )
  )
  (define (ex-f s e x)
         (case (content-port  x)
              ('in (case (state-phase s)
                    ('passive (set! (state-job-id s) (content-value x))
                              (hold-in 'busy (state-processing-times))  ;;;;;;;;;;;test features
                    )
                    ('busy (continue))
                    )
              )
        )
  )
  (define (in-f s)
    (case (state-phase s)
      ('busy (passivate))
  ) )
  (define (out-f s)
    (case (state-phase s)
      ('busy (set! (state-job-id s)(list (car (state-job-id s))(odding (car(cdr(state-job-id s))))))
            (make-content 'port 'out 'value (state-job-id s))
       )
    (else (make-content))
    )
  )
  (send pl set-ext-transfn ex-f)
  (send pl set-int-transfn in-f)
  (send pl set-outputfn out-f)
```

4.2. psm.m

```
(load "mbase/func.m")
  (make-pair atomic-models 'psm)
  (send psm def-state
          '(
            job-id         ;name of the processed job
            processing-time  ; processing time of this processor
           )
  )
  (send psm set-s
          (make-state 'sigma     'inf
                      'phase     'passive
                      'job-id    '()
                      'processing-time rnd-seed-tri
          )
  )
  (define (ex-f s e x)
        (case (content-port  x)
             ('in (case (state-phase s)
                  ('passive (set! (state-job-id s) (content-value x))
                           (hold-in 'busy (state-processing-time s))  ;;;;;;;;;;;test features
                  )
                  ('busy (continue))
                  )
             )
       )
  )
  (define (in-f s)
    (case (state-phase s)
      ('busy (passivate))
  ) )
  (define (out-f s)
    (case (state-phase s)
      ('busy (set! (state-job-id s)(list (car (state-job-id s))(odding (car(cdr(state-job-id s))))))
            (make-content 'port 'out 'value (state-job-id s))
       )
    (else (make-content))
    )
  )
  (send psm set-ext-transfn ex-f)
  (send psm set-int-transfn in-f)
  (send psm set-outputfn out-f)
```

4.3. pd.m

```
(load "mbase/func.m")
   (make-pair atomic-models 'pd)
   (send pd def-state
            '(
              job-id
              processing-time  ; processing time of this processor
             )
   )
   (send pd set-s
            (make-state 'sigma     'inf
                        'phase     'passive
                        'job-id    '()
                        'processing-time rnd-seed-tri

            )
   )
   (define (ex-f s e x)
          (case (content-port  x)
               ('in (case (state-phase s)
                    ('passive (set! (state-job-id s) (content-value x))
                              (hold-in 'busy (state-processing-time s))
                    )
                    ('busy (continue))
                    )
               )
         )
   )

   (define (in-f s)
     (case (state-phase s)
       ('busy (passivate))
   ) )
   (define (out-f s)
     (case (state-phase s)
       ('busy
              (set! (state-job-id s)(append (list (car (state-job-id s)))(split (car (cdr(state-job-id s))))))
         (make-content 'port 'out 'value (state-job-id s))
       )
     (else (make-content))
     )
   )
   (send pd set-ext-transfn ex-f)
   (send pd set-int-transfn in-f)
   (send pd set-outputfn out-f)
```

4.4. pc.m

```
(load "mbase/func.m")
   (make-pair atomic-models 'pc)
   (send pc def-state
            '(

               job-id
              processing-time
             )
   )
   (send pc set-s
            (make-state 'sigma     'inf
                        'phase     'passive
                        'job-id    '()
                        'processing-time rnd-seed-tri

            )
   )
   (define (ex-f s e x)
          (case (content-port  x)
               ('in (case (state-phase s)
                    ('passive (set! (state-job-id s) (content-value x))
                              (hold-in 'busy (state-processing-time s))
                    )
                    ('busy (continue))
                    )
               )
         )
   )
   (define (in-f s)
     (case (state-phase s)
       ('busy (passivate))
   ) )
   (define (out-f s)
     (case (state-phase s)
       ('busy
            (set! (state-job-id s)(list (car (state-job-id s))(merge (cdr(state-job-id s)))))
          (make-content 'port 'out 'value (state-job-id s))
       )
     (else (make-content))
     )
   )
   (send pc set-ext-transfn ex-f)
   (send pc set-int-transfn in-f)
   (send pc set-outputfn out-f)
```

4.5. genr.m

```
   (make-pair atomic-models 'genr)
   (send genr def-state '(inter-arrival-time))
  (send genr set-s (make-state 'sigma            0
                               'phase            'active
                               'inter-arrival-time 3
                   )
   )
   (define (ext-genr s e x)
     (case (content-port x)
           ('stop
              (passivate)  ;when receive stop signal passivate
           )
           (else (continue))
     )
   )
(define (int-genr s)
    (case (state-phase s)
          ('active
             (set! (state-sigma s) (state-inter-arrival-time s))
           )      ;;;reset sigma each time an internal transition occurs
                  ;;; Note: not really necessary for fixed inter-arrival
                  ;;; time
   ) )

   ; output the jobname (gensym) to port 'out

   (define (out-genr s)
     (case (state-phase s)
           ('active
             (make-content 'port 'out 'value
                (list (gensym 'g)
                       (list (random 100)(random 100)(random 100)(random 100)(random 100)
                             (random 100)(random 100)(random 100)(random 100)(random 100)
                             (random 100)(random 100)(random 100)(random 100)(random 100)
                       )
                )
              )
           )
           (else (make-content)
   )))

   (send genr set-int-transfn int-genr)
   (send genr set-ext-transfn ext-genr)
   (send genr set-outputfn out-genr)
```

4.6. transd.m

```
(define observation-interval 100)
(make-pair atomic-models 'transd)
(send transd def-state '(arrived-list  ;; all jobs which have arrived
                         solved-list   ;; all jobs which have been processed
                         clock         ;; local clock
                         total-ta)     ;; total turnaround time of processed
                                       ;; jobs
)
(send transd set-s (make-state
                         'sigma        observation-interval
                         'phase        'active
                         'arrived-list  '()
                         'solved-list   '()
                         'clock        0
                         'total-ta     0
)               )
(define (ext-t s e x)
     (let (
          (problem-id (car (content-value x)))
          (problem-content (content-value x))
          )
       (set! (state-clock s) (+ (state-clock s) e))
       (case (content-port x)
         ('ariv (set! (state-arrived-list s)
                      (cons (append problem-content (list (state-clock s)))
                            (state-arrived-list s)))
         )
         ('solved (let* (
                         (pair  (assoc problem-id (state-arrived-list s)))
                         (prob-arrival-time  (caddr pair))

                         (turn-around-time
                              (when prob-arrival-time
                              (- (state-clock s) prob-arrival-time))
                              )
                        )
                    (when  prob-arrival-time
                       (set! (state-total-ta s)
                          (+ (state-total-ta s) turn-around-time))
                       (set! (state-solved-list s)
                         (cons problem-content (state-solved-list s)))
                    )
                )
          )
         (else (bkpt "error: invalid input port name --> " (content-port x)))
       )
      )
      (continue)
)
(define (int-t s)
(case (state-phase s)
  ('active (passivate))
))
(define (out-t s)
  (case (state-phase s)
    ('active
       (let* (
            (log-file (open-output-file "log"))
            (avg-ta-time
              (if (NULL? (state-solved-list s))
                '()
                (/ (state-total-ta s) (length (state-solved-list s)))
              )
            )
            (thruput
             (if (= (state-clock s) 0)
               '()
               (/ (- (length (state-solved-list s)) 1)
                  (- (state-clock s) avg-ta-time))
             )
            )
          )
       (newline log-file)
       (display "The arrived list: " log-file)
       (display (state-arrived-list s) log-file)
       (newline log-file)
       (display "The solved list:  " log-file)
       (display (state-solved-list s) log-file)
           (newline log-file)
       (display "Avg. turnaround time: " log-file)
          (display avg-ta-time  log-file)
       (newline log-file)
       (display "ThruPut: " log-file)
          (display thruput log-file)
       (newline log-file)
       (close-output-port log-file)
       (make-content 'port 'out 'value (list avg-ta-time thruput))
     ) ;;let
    ) ;;active
    (else (make-content))
))
(send transd set-ext-transfn ext-t)
(send transd set-int-transfn int-t)
(send transd set-outputfn out-t)
```

4.7. mul-c.m

```
;Multi-sever Co-ordinator

(make-pair atomic-models 'mul-c)

(send mul-c def-state '(
                   spfour1-s
                   spfour2-s
                   spfour3-s
                   out-port
                   job
                   )
)
(send mul-c set-s    (make-state   'sigma 'inf
                                'phase 'passive
                                'spfour1-s 'passive
                                'spfour2-s 'passive
                                'spfour3-s 'passive
                                'out-port  '()
                                'job       '()
                    )
)
(define (ext-mc s e x)
   (set! (state-out-port s) '())  ;default , no port to be sent to
   (set! (state-job s)  (content-value x))
   (case  (content-port x)
         ('in
           (cond
             ( (equal? (state-spfour1-s s) 'passive)
                    (set! (state-out-port s) 'x1)
                    (set! (state-spfour1-s s) 'busy))
             ( (equal? (state-spfour2-s s) 'passive)
                    (set! (state-out-port s) 'x2)
                    (set! (state-spfour2-s s) 'busy))
             ( (equal? (state-spfour3-s s) 'passive)
                    (set! (state-out-port s) 'x3)
                    (set! (state-spfour3-s s) 'busy))
          ))
         ('y1 (set! (state-spfour1-s s) 'passive)
             (set! (state-out-port s) 'out)
         )
         ('y2 (set! (state-spfour2-s s) 'passive)
             (set! (state-out-port s) 'out)
         )
         ('y3 (set! (state-spfour3-s s) 'passive)
             (set! (state-out-port s) 'out)
         )
     )
     (hold-in 'busy 0)
)
(define (int-mc s)
   (case (state-phase s)
     ('busy
        (passivate)
)))
(define (out-mc s)
   (case (state-phase s)
     ('busy
        (case (state-out-port s)
           ((x1 x2 x3 out)
               (make-content 'port  (state-out-port s)
                             'value (state-job s))
           )
        (else (make-content))
))))
(send mul-c set-ext-transfn ext-mc)
(send mul-c set-int-transfn int-mc)
(send mul-c set-outputfn out-mc)
```

4.8. pip-c

```
(make-pair atomic-models 'pip-c)

(send pip-c def-state '(
                   out-port
                   job
                   )
)

(send pip-c set-s (make-state 'sigma 'inf
                         'phase 'passive
                         'out-port '()
                         'job   '()
                )
)

(define (ext-ppc s e x)
       (set! (state-out-port s) '())
       (set! (state-job s) (content-value x))
       (case (content-port x)
         ('in
            (set! (state-out-port s) 'x1)
         )
         ('y1 (set! (state-out-port s) 'x2))
         ('y2 (set! (state-out-port s) 'x3))
         ('y3 (set! (state-out-port s) 'out))
       )
       (hold-in 'busy 0)
)

(define (int-ppc s)
   (case (state-phase s)
     ('busy
        (passivate)
)))

(define (out-ppc s)
  (case (state-phase s)
    ('busy
      (case (state-out-port s)
        ((x1 x2 x3 out)
           (make-content 'port (state-out-port s)
                         'value (state-job s))
      )
      (else (make-content))
)))   )

(send pip-c set-ext-transfn ext-ppc)
(send pip-c set-int-transfn int-ppc)
(send pip-c set-outputfn out-ppc)
```

4.9. dc-c

```
   (make-pair atomic-models 'dc-c)
   (send dc-c def-state '(
                       p-cnt    ;; number of partial solutions received
                       out-port  ;; destination for next output
                       job-id
                       job-content
                       total-lst
                       ))
(send dc-c set-s (make-state 'sigma     'inf
                             'phase     'passive
                             'p-cnt     3
                             'out-port   '()
                             'job-id    '()
                             'job-content '()
                             'total-lst '()

   )              )
   (define (ex-dc s e x)
          (set! (state-out-port s) '())
          (set! (state-job-id s) (car (content-value x)))
          (set! (state-job-content s)(car (cdr (content-value x))))

          (case (content-port x)
            ('in
              (set! (state-out-port s) 'px)
            )

            ('py
               (set! (state-job-content s) (cdr (content-value x)))
               (set! (state-out-port s) 'xin)
            )
            ((y1 y2 y3)  (set! (state-total-lst s)(append (state-total-lst s)(list (state-job-content s))))
                         (set! (state-p-cnt s) (1+ (state-p-cnt s)))
                         (when (= (state-p-cnt s) 3)(set! (state-out-port s) 'cx)))
            ('cy   (set! (state-total-lst s) '())(set! (state-out-port s) 'out))
          ) ; end of case
          (hold-in 'busy 0)
   )
   (define (out-dc s)
      (case (state-phase s)
        ('busy
          (case (state-out-port s)
               ('xin    ;;; check whether the processors are all free
                  (if (= (state-p-cnt s) 3)
                    (begin
                      (set! (state-p-cnt s) 0)
                       (list
                        (make-content 'port 'x1  'value (list (state-job-id s)(car (state-job-content s))))
                        (make-content 'port 'x2  'value (list (state-job-id s)(cadr (state-job-content s))))
                        (make-content 'port 'x3  'value (list (state-job-id s)(caddr (state-job-content s))))
                       )); else send no partial jobs to processors
                    (make-content)))
               ((px out) (make-content 'port (state-out-port s) 'value (list (state-job-id s)(state-job-
content s))))
               ('cx
                    (make-content   'port  (state-out-port s)
                         'value (list (state-job-id s)(car(state-total-lst s))(cadr(state-total-lst
s))(caddr(state-total-lst s)))))
               (else (make-content)))   ) ) )
   (define (in-dc s)
     (case (state-phase s)
       ('busy
          (passivate)
   ) ) )
   (send dc-c set-ext-transfn ex-dc)
   (send dc-c set-int-transfn in-dc)
   (send dc-c set-outputfn out-dc)
```

4.10. mul-arch

```
;;;;;;;;;;;;mul-arch.m;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; The Multi-Server Architecture ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(load-from model-base_directory pl.m)
(load-from model-base directory mul-c.m)

;; make three copies from original p processor and copy its initial state

(send pl make-new 'pl1)
(send pl1 copy-state pl)

(send pl make-new 'pl2)
(send pl2 copy-state pl)

(send pl make-new 'pl3)
(send pl3 copy-state pl)

;;now couple them to the multi-server

(make-pair digraph-models 'mul-arch)

;;composition tree

(send mul-arch build-composition-tree
            mul-arch
            (list mul-c pl1 pl2 pl3))

;;influence digraph

(send mul-arch set-inf-dig (list (list mul-c pl1 pl2 pl3)
                              (list pl1 mul-c)
                              (list pl2 mul-c)
                              (list pl3 mul-c)))

;;internal coupling between processors and co-ordinator

(send mul-arch set-int-coup mul-c pl1 (list (cons 'x1 'in)))
(send mul-arch set-int-coup pl1 mul-c (list (cons 'out 'y1)))
(send mul-arch set-int-coup mul-c pl2 (list (cons 'x2 'in)))
(send mul-arch set-int-coup pl2 mul-c (list (cons 'out 'y2)))
(send mul-arch set-int-coup mul-c pl3 (list (cons 'x3 'in)))
(send mul-arch set-int-coup pl3 mul-c (list (cons 'out 'y3)))

;; external-input coupling

(send mul-arch set-ext-inp-coup mul-c (list (cons 'in 'in)))

;; external -output coupling

(send mul-arch set-ext-out-coup mul-c (list (cons 'out 'out)))

(define (sel-mul slst)
    (cond ((member pl1 slst) pl1)
          ((member pl2 slst) pl2)
          ((member pl3 slst) pl3)
          ((member mul-c slst) mul-c)
   ))

(send mul-arch set-selectfn sel-mul)

 (send mul-arch set-priority (list pl1 pl2 pl3 mul-c))
```

4.11. pip-arch

```
;;;;;;;;;;;;pip-arch.m;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; The Pipe-line Architecture    ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(load-from model-base directory psm.m)
(load-from model-base_directory pip-c.m)

;; make three copies from original p processor and copy its initial state

(send psm make-new 'psm1)
(send psm1 copy-state psm)

(send psm make-new 'psm2)
(send psm2 copy-state psm)

(send psm make-new 'psm3)
(send psm3 copy-state psm)

;;now couple them to the pipeline co-ordinator

(make-pair digraph-models 'pip-arch)

;;composition tree

(send pip-arch build-composition-tree
            pip-arch
            (list pip-c psm1 psm2 psm3))

;;influence digraph

(send pip-arch set-inf-dig (list (list pip-c psm1 psm2 psm3)
                                 (list psm1 pip-c)
                                 (list psm2 pip-c)
                                 (list psm3 pip-c)))

;;internal coupling between processors and co-ordinator

(send pip-arch set-int-coup pip-c psm1 (list (cons 'x1 'in)))
(send pip-arch set-int-coup psm1 pip-c (list (cons 'out 'y1)))
(send pip-arch set-int-coup pip-c psm2 (list (cons 'x2 'in)))
(send pip-arch set-int-coup psm2 pip-c (list (cons 'out 'y2)))
(send pip-arch set-int-coup pip-c psm3 (list (cons 'x3 'in)))
(send pip-arch set-int-coup psm3 pip-c (list (cons 'out 'y3)))

;; external-input coupling

(send pip-arch set-ext-inp-coup pip-c (list (cons 'in 'in)))

;; external-output coupling

(send pip-arch set-ext-out-coup pip-c (list (cons 'out 'out)))
(define (sel-pip slst)
    (cond ((member psm1 slst) psm1)
          ((member psm2 slst) psm2)
          ((member psm3 slst) psm3)
          ((member pip-c slst) pip-c)
   ))

(send pip-arch set-selectfn sel-pip)

;; equivalently

 (send pip-arch set-priority (list psm1 psm2 psm3 pip-c))
```

4.12. dc-arch

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;Contents of file: dc-arch.m ;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; The Module of Divide and Conquer Architecture
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;-----------------------------------------------------------------


;; get one base processor and one co-ordinator
(load-from model-base_directory pd.m)
(load-from model-base_directory pc.m)
(load-from model-base_directory psm.m)
(load-from model-base_directory dc-c.m)

(send pd make-new 'p&div)

; and the post-compiler

(send pc make-new 'p&cmpl)

(send psm make-new 'psm1)
(send psm make-new 'psm2)
(send psm make-new 'psm3)

;; now couple them together

(make-pair digraph-models 'dc-arch)


(send dc-arch build-composition-tree
                dc-arch
                (list dc-c p&div psm1 psm2 psm3 p&cmpl)
)
                                        ;;p&div: partition  processor
                                        ;;p&cmpl: compiler

;; influence digraph

(send dc-arch set-inf-dig (list (list dc-c p&div psm1 psm2 psm3 p&cmpl)
                                (list p&div dc-c)
                                (list psm1 dc-c)
                                (list psm2 dc-c)
                                (list psm3 dc-c)
                                (list p&cmpl dc-c)))

;; internal coupling

(send dc-arch set-int-coup dc-c p&div (list (cons 'px  'in)))
(send dc-arch set-int-coup p&div dc-c (list (cons 'out 'py)))
(send dc-arch set-int-coup dc-c psm1 (list (cons 'x1  'in)))
(send dc-arch set-int-coup psm1 dc-c (list (cons 'out 'y1)))
(send dc-arch set-int-coup dc-c psm2 (list (cons 'x2  'in)))
(send dc-arch set-int-coup psm2 dc-c (list (cons 'out 'y2)))
(send dc-arch set-int-coup dc-c psm3 (list (cons 'x3  'in)))
(send dc-arch set-int-coup psm3 dc-c (list (cons 'out 'y3)))
(send dc-arch set-int-coup dc-c p&cmpl (list (cons 'cx  'in)))
(send dc-arch set-int-coup p&cmpl dc-c (list (cons 'out 'cy)))

;; external-internal coupling

(send dc-arch set-ext-inp-coup dc-c (list (cons 'in 'in)))

;; internal-external coupling

(send dc-arch set-ext-out-coup dc-c (list (cons 'out 'out)))

(define (sel-dcc slst)
        (cond   ((member p&cmpl slst) p&cmpl)
                ((member psm1  slst) psm1)
                ((member psm2  slst) psm2)
                ((member psm3  slst) psm3)
                ((member p&div  slst) p&div)
                ((member dc-c slst) dc-c)
        )
)

(send dc-arch set-selectfn sel-dcc)

(send dc-arch set-priority (list p&cmpl psm1 psm2 psm3 p&div dc-c))
```

4.13. ef

```
(load-from  model-base_directory genr.m)
(load-from  model-base_directory transd.m)

;; couple them in a digraph-model

(make-pair digraph-models 'ef)
(send ef build-composition-tree ef (list genr transd))
(send ef set-inf-dig (list (list genr transd) (list transd genr)))

;; the connection from transducer to generator is made
;; for the termination of experiment

(send ef set-int-coup transd genr (list (cons 'out 'stop)))

(send ef set-int-coup genr transd (list (cons 'out 'ariv)))
(send ef set-ext-inp-coup transd (list (cons 'in 'solved)))
(send ef set-ext-out-coup genr (list (cons 'out 'out)))
(send ef set-ext-out-coup transd (list (cons 'out 'result)))
```

4.14. ef-pl

```
;; load the components

(load-from  model-base_directory pl.m)

(load-from  model-base_directory ef.m)

;; couple the experimental frame with the processor by p-ef

(make-pair digraph-models 'ef-pl)

;; the composition tree

(send ef-pl build-composition-tree ef-pl (list pl ef))

;; the influence digraph
(send ef-pl set-inf-dig (list (list pl ef)
                              (list ef pl)
)                          )

;; internal coupling
(send ef-pl set-int-coup pl ef (list (cons 'out 'in)))
(send ef-pl set-int-coup ef pl (list (cons 'out 'in)))

(send ef-pl set-ext-out-coup ef  (list (cons 'result 'out)))


;; define the select function to avoid  collision when the job
;; arrives at the time the processor finishes: processor first
;; then generator

(define (sel-p slst)
   (cond ((member pl slst) pl)
         ((member ef slst) ef)
)  )

(send ef-pl set-selectfn sel-p)

;; equivalently

 (send ef-pl set-priority (list pl ef))

;; is shorter and preferable when using flat-devs and deep-devs
;; the final touch, attach a root co-ordinator

(mk-ent root-co-ordinators r)

;; initialize it with the co-ordinator for ef-p

(initialize r c:ef-pl)

;; start a simulation run

(restart r)
```

4.15. ef-mul

```
;; load the components

(load-from  model-base_directory mul-arch.m)

(load-from  model-base_directory ef.m)

;; couple the experimental frame with the processor by p-ef

(make-pair digraph-models 'ef-mul)

;; the composition tree

(send ef-mul build-composition-tree ef-mul (list mul-arch ef)

;; the influence digraph
(send ef-mul set-inf-dig (list (list mul-arch ef)
                               (list ef mul-arch)
)                          )

;; internal coupling
(send ef-mul set-int-coup mul-arch ef (list (cons 'out 'in)))
(send ef-mul set-int-coup ef mul-arch (list (cons 'out 'in)))

(send ef-mul set-ext-out-coup ef  (list (cons 'result 'out)))


(define (sel-mul slst)
   (cond ((member mul-arch slst) mul-arch)
         ((member ef slst) ef)
)  )

(send ef-mul set-selectfn sel-mul)
(send ef-mul set-priority (list mul-arch ef))



(mk-ent root-co-ordinators r)
 (initialize r c:ef-mul)
 (restart r)
```

4.16. ef-pip

```
(load-from  model-base_directory pip-arch.m)
(load-from  model-base_directory ef4pip.m)


(make-pair digraph-models 'ef-pip)

;; the composition tree

(send ef-pip build-composition-tree ef-pip (list pip-arch ef))

;; the influence digraph
(send ef-pip set-inf-dig (list (list pip-arch ef)
                               (list ef pip-arch)
)                        )

;; internal coupling
(send ef-pip set-int-coup pip-arch ef (list (cons 'out 'in)))
(send ef-pip set-int-coup ef pip-arch (list (cons 'out 'in)))

(send ef-pip set-ext-out-coup ef  (list (cons 'result 'out)))

(define (sel-pip slst)
   (cond ((member pip-arch slst) pip-arch)
         ((member ef slst) ef)
)  )

(send ef-pip set-selectfn sel-pip)

;; equivalently
(send ef-pip set-priority (list pip-arch ef))

(mk-ent root-co-ordinators r)
(initialize r c:ef-pip)
(restart r)
```

4.17. ef-dc

```
(load-from  model-base_directory dc-arch.m)
(load-from  model-base_directory ef.m)

(make-pair digraph-models 'ef-dc)
(send ef-dc build-composition-tree ef-dc (list dc-arch ef))

;; the influence digraph
(send ef-dc set-inf-dig (list (list dc-arch ef)
                              (list ef dc-arch)
)                       )

;; internal coupling
(send ef-dc set-int-coup dc-arch ef (list (cons 'out 'in)))
(send ef-dc set-int-coup ef dc-arch (list (cons 'out 'in)))

(send ef-dc set-ext-out-coup ef  (list (cons 'result 'out)))

(define (sel-dc slst)
   (cond ((member dc-arch slst) dc-arch)
         ((member ef slst) ef)
)  )

(send ef-dc set-selectfn sel-dc)


(send ef-dc set-priority (list dc-arch ef))

(mk-ent root-co-ordinators r)
(initialize r c:ef-dc)
(restart r)
```

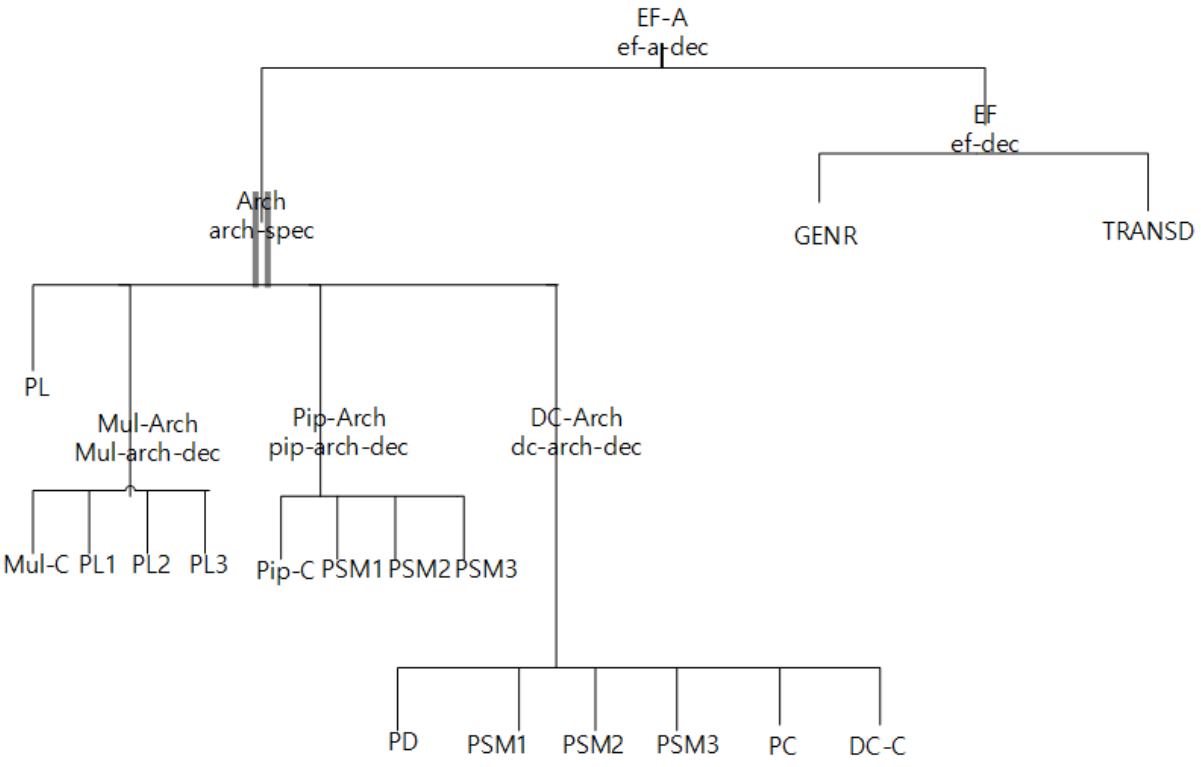
5. SES

5.1. codes

```
   (make-entstr 'ef-a)
   (add-item e:ef-a asp 'ef-a-dec)
   (set-current-item e:ef-a 'ef-a-dec)
   (add-item e:ef-a ent 'arch)
   (add-item e:ef-a ent 'ef)
   ;-------- coupling -----------------------------------
   (add-couple e:ef-a 'arch 'ef 'out 'in)
   (add-couple e:ef-a 'ef 'arch 'out 'in)
   (add-couple e:ef-a 'ef 'ef-a 'result 'out)
   ;-----------------------------------------------------
   (add-priority e:ef-a '(arch ef))
   (set-current-item e:ef-a 'arch)
   (add-item e:ef-a spec 'arch-spec)
   (set-current-item e:ef-a 'arch-spec)

   (add-item e:ef-a ent 'pl)
   (add-item e:ef-a ent 'pip)
   (add-item e:ef-a ent 'mul)
   (add-item e:ef-a ent 'd&c)
  ;-------- pip ----------------------------------
   (set-current-item e:ef-a 'pip)
   (add-item e:ef-a asp 'pip-dec)
   (set-current-item e:ef-a 'pip-dec)
   (add-item e:ef-a ent 'pip-c)
   (add-item e:ef-a ent 'psm1)
   (add-item e:ef-a ent 'psm2)
   (add-item e:ef-a ent 'psm3)
   (add-couple e:ef-a 'pip 'pip-c 'in 'in)
   (add-couple e:ef-a 'pip-c 'pip 'out 'out)
   (add-couple e:ef-a 'pip-c 'psm1 'x1 'in)
   (add-couple e:ef-a 'pip-c 'psm2 'x2 'in)
   (add-couple e:ef-a 'pip-c 'psm3 'x3 'in)
   (add-couple e:ef-a 'psm1 'pip-c 'out 'y1)
   (add-couple e:ef-a 'psm2 'pip-c 'out 'y2)
   (add-couple e:ef-a 'psm3 'pip-c 'out 'y3)
   (add-priority e:ef-a '(psm1 psm2 psm3 pip-c))
;;; -- mul
   (set-current-item e:ef-a 'mul)
   (add-item e:ef-a asp 'mul-dec)
   (set-current-item e:ef-a 'mul-dec)
   (add-item e:ef-a ent 'mul-c)
   (add-item e:ef-a ent 'pl1)
   (add-item e:ef-a ent 'pl2)
   (add-item e:ef-a ent 'pl3)
   (add-couple e:ef-a 'mul 'mul-c 'in 'in)
   (add-couple e:ef-a 'mul-c 'mul 'out 'out)
   (add-couple e:ef-a 'mul-c 'pl1 'x1 'in)
   (add-couple e:ef-a 'mul-c 'pl2 'x2 'in)
   (add-couple e:ef-a 'mul-c 'pl3 'x3 'in)
   (add-couple e:ef-a 'pl1 'mul-c 'out 'y1)
   (add-couple e:ef-a 'pl2 'mul-c 'out 'y2)
   (add-couple e:ef-a 'pl3 'mul-c 'out 'y3)
   (add-priority e:ef-a '(pl1 pl2 pl3 mul-c))
```

```
;;; -- d&c
   (set-current-item e:ef-a 'd&c)
   (add-item e:ef-a asp 'd&c-dec)
   (set-current-item e:ef-a 'd&c-dec)
   (add-item e:ef-a ent 'dc-c)
   (add-item e:ef-a ent 'psm1)
   (add-item e:ef-a ent 'psm2)
   (add-item e:ef-a ent 'psm3)
   (add-item e:ef-a ent 'p&div)
   (add-item e:ef-a ent 'p&cmpl)
   (add-couple e:ef-a 'd&c 'dc-c 'in 'in)
   (add-couple e:ef-a 'dc-c 'd&c 'out 'out)
   (add-couple e:ef-a 'dc-c 'psm1 'x1 'in)
   (add-couple e:ef-a 'dc-c 'psm2 'x2 'in)
   (add-couple e:ef-a 'dc-c 'psm3 'x3 'in)
   (add-couple e:ef-a 'dc-c 'p&div 'px 'in)
   (add-couple e:ef-a 'dc-c 'p&cmpl 'cx 'in)
   (add-couple e:ef-a 'psm1 'dc-c 'out 'y1)
   (add-couple e:ef-a 'psm2 'dc-c 'out 'y2)
   (add-couple e:ef-a 'psm3 'dc-c 'out 'y3)
   (add-couple e:ef-a 'p&div 'dc-c 'out 'py)
   (add-couple e:ef-a 'p&cmpl 'dc-c 'out 'cy)
   (add-priority e:ef-a '(p&cmpl p3 p2 p1 p&div dc-c))

   (set-current-item e:ef-a 'ef)
   (add-item e:ef-a asp 'ef-dec)

   ;; experimental frame consists of generator and transducer
   (set-current-item e:ef-a 'ef-dec)
   (add-item e:ef-a ent 'transd)
   (add-item e:ef-a ent 'genr)

   ;-------- coupling ------------------------------------
   (add-couple e:ef-a 'ef 'transd 'in 'solved)
   (add-couple e:ef-a 'transd 'ef 'out 'result)
   (add-couple e:ef-a 'transd 'genr 'out 'stop)
   (add-couple e:ef-a 'genr 'ef 'out 'out)
   (add-couple e:ef-a 'genr 'transd 'out 'ariv)

   ;; save the entity structure
   (save-en e:ef-a)
```

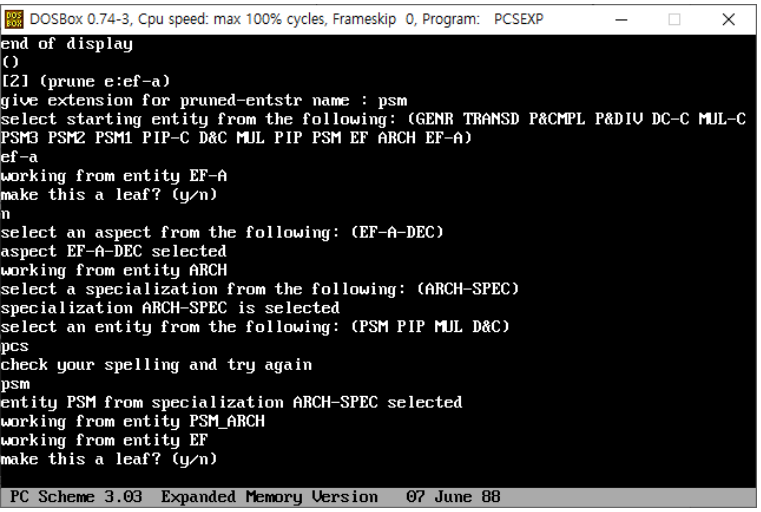
5.2. diagram

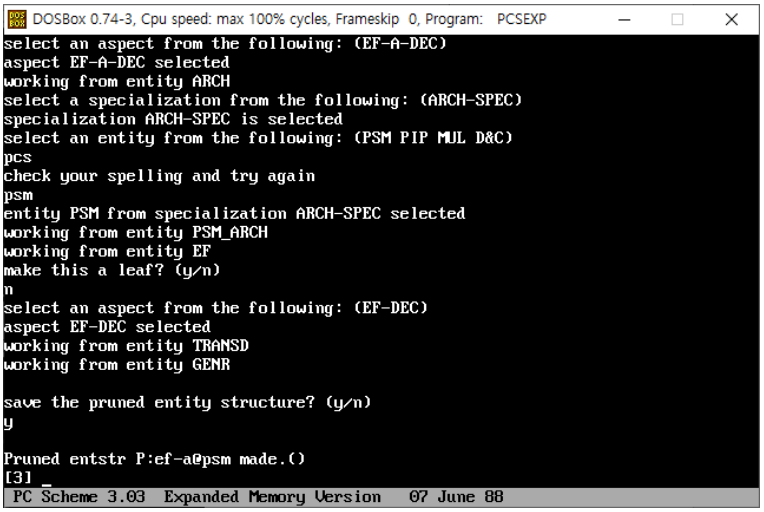


5.3. pl

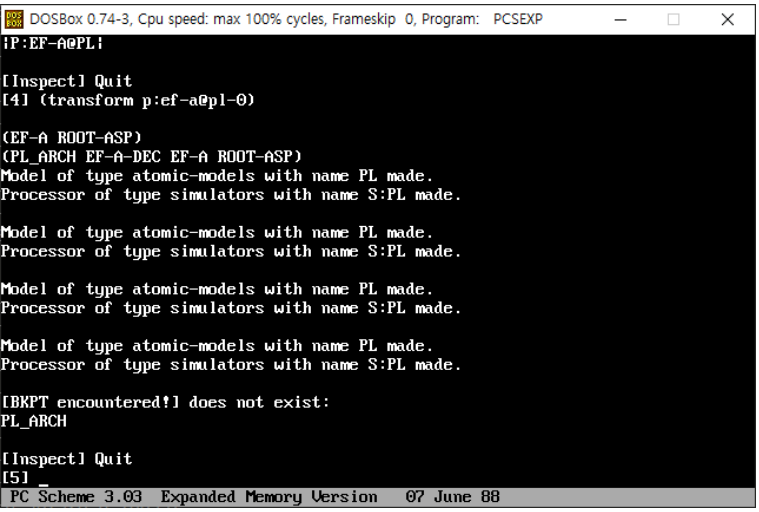
transform 시에 Pl.m atomic model 을 생성하지 못하는 오류가 있어서 psm.m 의 process time 을 바꾸어 사용하여 실행하였습니다.

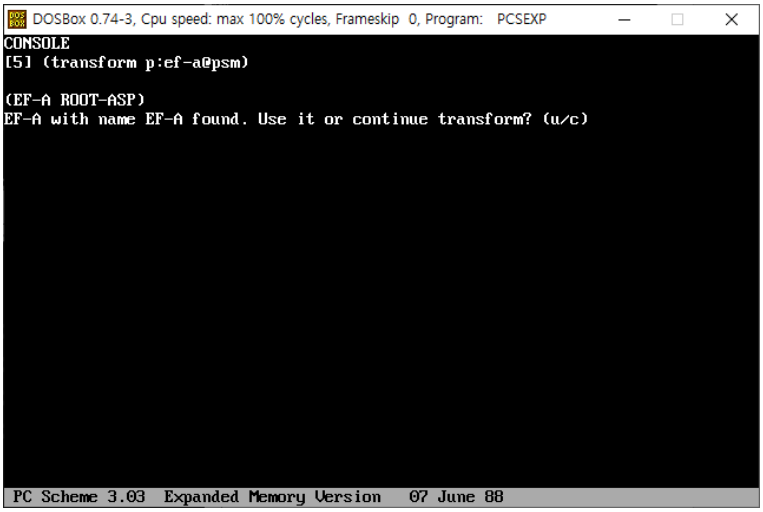
5.3.1. prune





5.3.2. transform

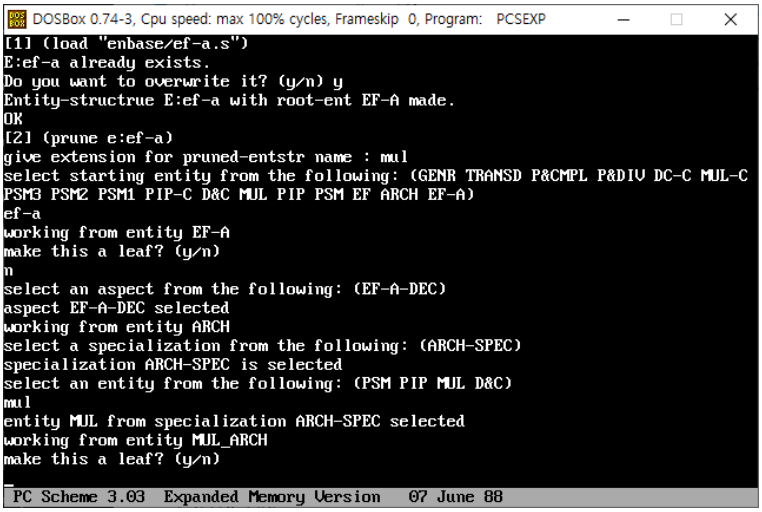


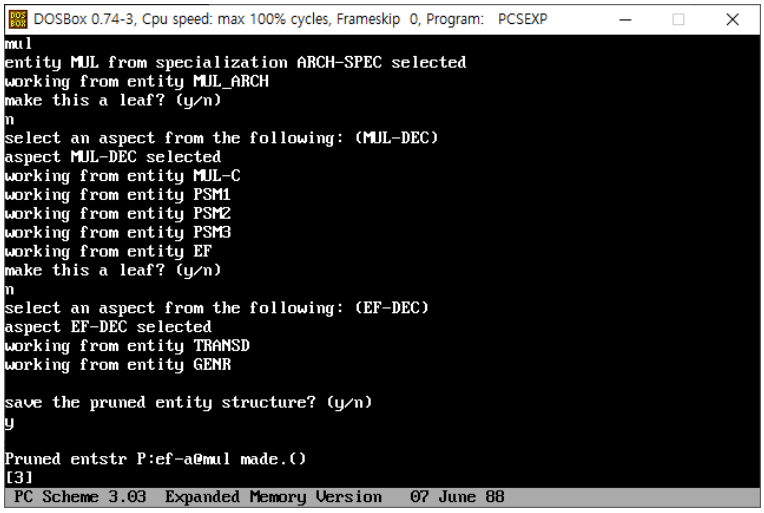


5.4. multiserver

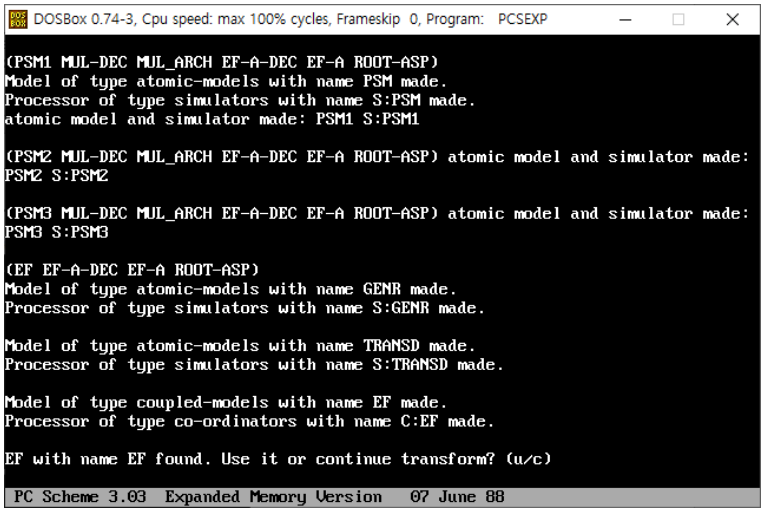
앞선 경우와 마찬가지로 pl 모델을 로드하지 못하는 오류가 있어 psm 모델의 processing time 만 대체하여 실행하였습니다.

5.4.1. prune



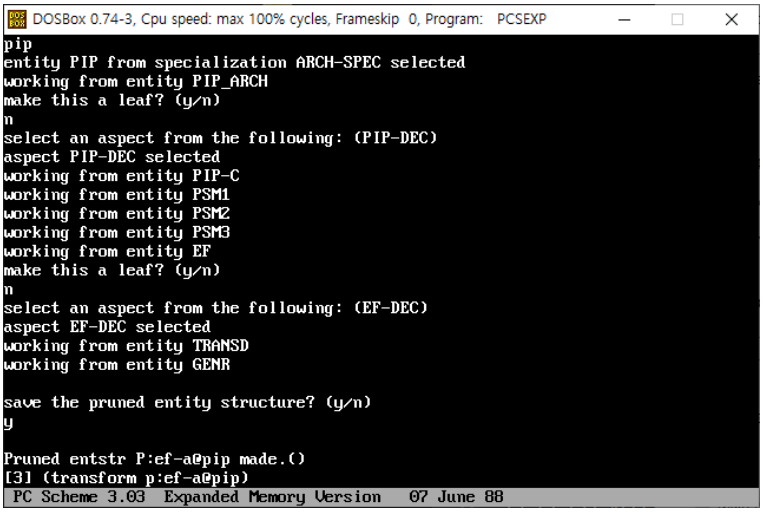


5.4.2. transform

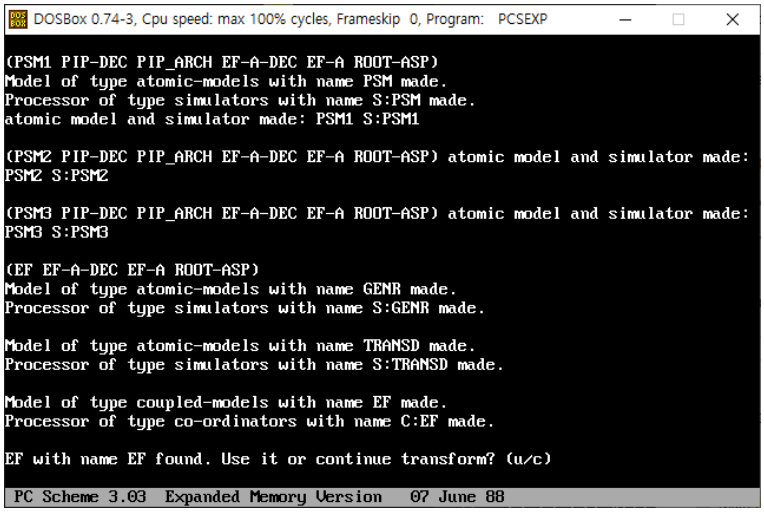


5.5. pipeline

5.5.1. prune

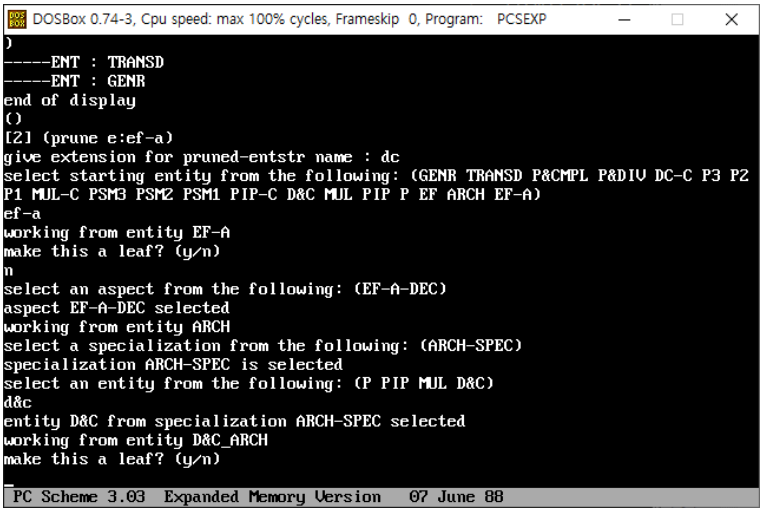


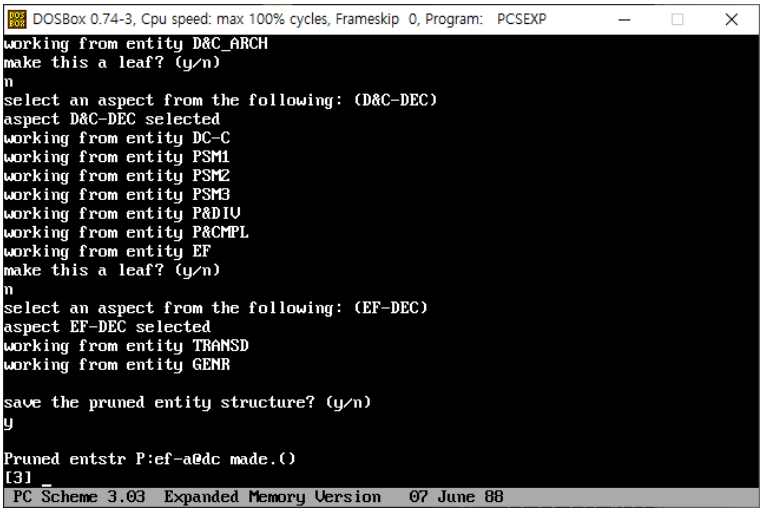
5.5.2. transform



5.6. divide & conquer

5.6.1. prune





5.6.2. transform