



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 1

по дисциплине «Системы массового обслуживания»

ВАРИАНТ 6

Тема: **Одноканальные системы массового обслуживания**

Выполнил:
Студент 4-го курса
Едренников Д.А.

Группа: КМБО-01-20

МОСКВА – 2023

Задание

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени t характеризуется числом заявок, находящихся в системе. События в развитии систем связаны либо с поступлением в неё новых заявок, либо с окончанием обслуживания прибором заявки.

В системах с одним прибором с отказами имеется 2 состояния: 0 – в системе 0 заявок (прибор свободен), 1 – в системе 1 заявка (прибор занят). В этом случае события могут быть трёх типов: 1 – появление в СМО новой заявки, которая сразу же принимается прибором на обслуживание (до этого прибор был свободен и СМО переходит из состояния 0 в состояние 1), 2 – появление в СМО новой заявки, которая получает отказ в обслуживании (прибор занят, при этом СМО остается в состоянии 1), 3 – завершение обслуживания заявки прибором (при этом СМО переходит из состояния 1 в состояние 0).

В системах с одним прибором и бесконечной очередью в состоянии $C = 0$ в системе 0 заявок (прибор свободен), в состоянии $C = k$ в системе k заявок (прибор занят, одна заявка обслуживается, $k = 1$ заявка находится в очереди типа FIFO). События могут быть двух типов: 1 – появление в СМО новой заявки, 2 – завершение обслуживания заявки прибором (при этом прибор освобождается, и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в прибор). Если при появлении в СМО новой заявки прибор свободен, то она сразу же принимается на обслуживание прибором, в противном случае заявка становится в очередь.

1. Одноканальная СМО с отказами ($D|M|1|0$)

Дано:

- время между приходом заявок ΔT_3 (заданная постоянная величина);
- параметр μ показательного распределения времени обслуживания заявки прибором.

Предполагается, что в начальный момент времени $t = 0$ в СМО нет заявок, т.е. состояние системы 0, и через заданное время ΔT_3 в СМО поступает первая заявка (произойдет событие с номером 1). Момент наступления первого события (типа 1) равен $t_{\text{cob}}(1) = \Delta T_3$. После события 1 СМО находится в состоянии 1, в котором она будет оставаться время $t_{\text{обсл}}(1)$, определяемое в соответствии с показательным законом распределения с параметром μ .

2. Одноканальная СМО с бесконечной очередью ($M|D|1$)

Дано:

- среднее число заявок λ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром λ);

- время обслуживания заявки прибором $T_{об}$ (заданная постоянная величина).

Предполагается, что в начальный момент времени $t = 0$ СМО находится в состоянии 0 и в этот момент определяется время поступления в СМО первой заявки $t_3(1)$ в соответствии с показательным законом распределения с параметром λ .

3. Одноканальная СМО с бесконечной очередью (M|M|1)

Дано:

- среднее число заявок λ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром λ);
- параметр μ показательного распределения времени обслуживания заявки прибором.

Предполагается, что в начальный момент времени $t = 0$ система находится в состоянии 0 и в этот момент определяется время поступления в систему первой заявки $t_3(1)$ в соответствии с показательным законом распределения с параметром λ , а в момент поступления каждой заявки на обслуживание в прибор определяется время её обслуживания $t_{обсл}(1)$ в соответствии с показательным законом распределения с параметром μ .

Требуется:

1. Провести моделирование первых 100 событий в развитии каждой системы.
2. Составить таблицу 1 с данными о событиях:
 - номер события l ;
 - момент наступления события $t_{собр}(l)$;
 - тип события $Type(l)$;
 - состояние СМО $S(l)$ после события l ;
 - оставшееся время $t_{ост}(l)$ обслуживания прибором заявки после события l (если после события прибор свободен, то $t_{ост}(l) = -1$);
 - время ожидания $t_{ожз}(l)$, через которое после события l в СМО появится новая заявка;
 - номер заявки $j(l)$, участвующей в событии l .
3. Для СМО (D|M|1|0) составить таблицу 2 со следующими данными о всех поступивших заявках:
 - номер заявки j ;
 - момент $t_3(j)$ появления заявки j в СМО;
 - время $t_{обсл}(j)$ обслуживания прибором заявки j ;
 - момент $t_{коб}(j)$ окончания обслуживания заявки j и выхода её из СМО.

Если в момент появления заявки j в СМО прибор был занят, и заявка получила отказ в обслуживании, то $t_{обсл}(j) = 0$ и $t_{коб}(j) = t_3(j)$.

Для СМО (M|D|1) и (M|M|1) составить таблицу 2 со следующими данными о всех поступивших заявках:

- номер заявки j ;
- момент $t_3(j)$ появления заявки j в СМО;
- номер места в очереди $q(j)$, на которое попала заявка j (если заявка сразу начала обслуживаться, то номер места в очереди $q(j) = 0$);
- время пребывания заявки в очереди $t_{оч}(j)$;
- момент начала обслуживания заявки $t_{ноб}(j)$;
- время $t_{обсл}(j)$ обслуживания прибором заявки j ;
- момент $t_{коб}(j)$ окончания обслуживания заявки j и выхода её из СМО.

Если в момент появления заявки j в СМО прибор занят, и она становится в очередь, то в таблицу 2 временно заносится: $t_{оч}(j) = t_{ноб}(j) = t_{обсл}(j) = t_{коб}(j) = -1$.

Настоящие значения заносятся позже по мере их определения.

4. Для СМО (D|M|1|0) составить таблицу 3 с данными о состояниях вида:

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	$R_0(100)$	$v_0(100)$	$T_0(100)$	$\Delta_0(100)$
1	$R_1(100)$	$v_1(100)$	$T_1(100)$	$\Delta_1(100)$
	$\sum_i R_i(100)$	$\sum_i v_i(100)$	$\sum_i T_i(100)$	$\sum_i \Delta_i(100)$

а для СМО (M|D|1) и (M|M|1) таблицу 3 с данными о состояниях вида:

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	$R_0(100)$	$v_0(100)$	$T_0(100)$	$\Delta_0(100)$
1	$R_1(100)$	$v_1(100)$	$T_1(100)$	$\Delta_1(100)$
2	$R_2(100)$	$v_2(100)$	$T_2(100)$	$\Delta_2(100)$
....
	$\sum_i R_i(100)$	$\sum_i v_i(100)$	$\sum_i T_i(100)$	$\sum_i \Delta_i(100)$

Где:

$R_i(100)$ – число попаданий СМО в состояние i в событиях с 1-го по 100 ;

$v_i(100) = \frac{R_i(100)}{100}$ – относительная частота попадания СМО в состояние i в событиях с 1-го по 100 ;

$T_i(100)$ – общее время пребывания СМО в состоянии i на интервале $[0, t_{коб}(100)]$;

$\Delta_i(100) = \frac{T_i(100)}{t_{\text{cob}}(100)}$ – доля времени пребывания СМО в состоянии i на интервале $[0, t_{\text{cob}}(100)]$;

5. Для СМО (D|M|1|0) найти:

- число заявок $J(100)$, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$;
- число $JF(100)$ полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$;
- число $JL(100)$ отклонённых заявок на интервале $[0, t_{\text{cob}}(100)]$;
- долю отклонённых заявок в общем числе поступивших в СМО заявок на интервале $[0, t_{\text{cob}}(100)]$;
- коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ (отношение времени простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ к $t_{\text{cob}}(100)$).

Для СМО (M|D|1) и (M|M|1) найти:

- число заявок $J(100)$, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$;
- число $JF(100)$ полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$;
- среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\bar{z} = \frac{1}{100} \sum_{l=1}^{100} z(l)$, где $z(l)$ – число заявок в СМО после события l ;
- среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\overline{t_{\text{оч}}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j)$;
- среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\overline{t_{\text{СМО}}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_3(j)]$;
- коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ (отношение времени простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ к $t_{\text{cob}}(100)$).

Вывод результатов проводить с округлением до 0,00001.

Краткие теоретические сведения

Система массового обслуживания (СМО) – это математическая модель систем, предназначенных для обслуживания заявок (требований, запросов, клиентов, заказчиков...), поступающих в нее, как правило, в случайные моменты времени.

Устройства или субъекты, занимающиеся обслуживанием – приборы, службы, аппараты, каналы и т.д.

Основные характеристики эффективности функционирования СМО:

1. Показатели эффективности использования СМО: среднее число заявок, которое может обслужить СМО за единицу времени; средняя продолжительность периода занятости СМО; коэффициент использования СМО и т.п.
2. Показатели качества обслуживания заявок: среднее время ожидания заявки в очереди, среднее время пребывания заявки в СМО, вероятность отказа требованию в обслуживании без ожидания, вероятность того, что поступившая заявка будет принята к обслуживанию сразу, закон распределения времени ожидания заявки в очереди, закон распределения пребывания заявки в СМО, средняя длина очереди, среднее число заявок, находящихся в системе, и т.п.
3. Показатели экономической и финансовой эффективности функционирования СМО: средние расходы на обеспечение работы СМО в определенный период времени (неделя, месяц, год, ...), средний доход приносимый СМО за этот период времени и т.п.

Одноканальное СМО - система массового обслуживания, в которой все задачи, поступающие в систему, обслуживаются в одном канале.

Для СМО (M|M|1):

Стационарные вероятности состояний:

$$\begin{cases} r_0 = 1 - \rho \\ r_k = \rho^k (1 - \rho) \\ \rho = \frac{\lambda}{\mu} \end{cases}$$

Средняя длина очереди:

$$\bar{z} = \frac{\rho}{1-\rho}$$

Среднее время пребывания заявок в очереди:

$$\overline{t_{оч}} = \frac{\bar{q}}{\lambda}$$

Среднее время пребывания заявок в СМО:

$$\overline{t_{СМО}} = \frac{\bar{z}}{\lambda}$$

Используемые функции из языка python:

expon.rvs(scale=1 / μ , size=100) – генерации случайных значений из экспоненциального распределения.

Результаты расчетов

СМО (D|M|1|0):

Вариант 6. $\Delta T_3 = 0,784$; $\mu = 1,254$

Таблица №1

l	$t_{собр}(l)$	Type(l)	состояние СМО C(l) после события l	$t_{ост}(l)$	$t_{ожз}(l)$	j(l)
1	0.784	1	1	0.05963	0.784	1
2	0.84363	3	0	-1.	0.72437	1
3	1.568	1	1	0.72127	0.784	2
4	2.28927	3	0	-1.	0.06273	2
5	2.352	1	1	2.0171	0.784	3
6	3.136	2	1	1.2331	0.784	4
7	3.92	2	1	0.4491	0.784	5
8	4.3691	3	0	-1.	0.3349	3
9	4.704	1	1	0.32372	0.784	6
10	5.02772	3	0	-1.	0.46028	6
11	5.488	1	1	0.4493	0.784	7
12	5.9373	3	0	-1.	0.3347	7
13	6.272	1	1	1.38899	0.784	8
14	7.056	2	1	0.60499	0.784	9
15	7.66099	3	0	-1.	0.17901	8
16	7.84	1	1	0.29	0.784	10
17	8.13	3	0	-1.	0.494	10
18	8.624	1	1	2.82485	0.784	11
19	9.408	2	1	2.04085	0.784	12
20	10.192	2	1	1.25685	0.784	13
21	10.976	2	1	0.47285	0.784	14
22	11.44885	3	0	-1.	0.31115	11

23	11.76	1	1	0.00489	0.784	15
24	11.76489	3	0	-1.	0.77911	15
25	12.544	1	1	0.80921	0.784	16
26	13.328	2	1	0.02521	0.784	17
27	13.35321	3	0	-1.	0.75879	16
28	14.112	1	1	0.53381	0.784	18
29	14.64581	3	0	-1.	0.25019	18
30	14.896	1	1	0.33802	0.784	19
31	15.23402	3	0	-1.	0.44598	19
32	15.68	1	1	0.82055	0.784	20
33	16.464	2	1	0.03655	0.784	21
34	16.50055	3	0	-1.	0.74745	20
35	17.248	1	1	0.25591	0.784	22
36	17.50391	3	0	-1.	0.52809	22
37	18.032	1	1	1.18827	0.784	23
38	18.816	2	1	0.40427	0.784	24
39	19.22027	3	0	-1.	0.37973	23
40	19.6	1	1	0.26642	0.784	25
41	19.86642	3	0	-1.	0.51758	25
42	20.384	1	1	1.74246	0.784	26
43	21.168	2	1	0.95846	0.784	27
44	21.952	2	1	0.17446	0.784	28
45	22.12646	3	0	-1.	0.60954	26
46	22.736	1	1	0.07253	0.784	29
47	22.80853	3	0	-1.	0.71147	29
48	23.52	1	1	0.21296	0.784	30
49	23.73296	3	0	-1.	0.57104	30
50	24.304	1	1	1.53165	0.784	31
51	25.088	2	1	0.74765	0.784	32
52	25.83565	3	0	-1.	0.03635	31
53	25.872	1	1	0.00628	0.784	33
54	25.87828	3	0	-1.	0.77772	33
55	26.656	1	1	1.17976	0.784	34
56	27.44	2	1	0.39576	0.784	35
57	27.83576	3	0	-1.	0.38824	34
58	28.224	1	1	1.00696	0.784	36
59	29.008	2	1	0.22296	0.784	37
60	29.23096	3	0	-1.	0.56104	36
61	29.792	1	1	1.02886	0.784	38
62	30.576	2	1	0.24486	0.784	39
63	30.82086	3	0	-1.	0.53914	38
64	31.36	1	1	1.18124	0.784	40
65	32.144	2	1	0.39724	0.784	41

66	32.54124	3	0	-1.	0.38676	40
67	32.928	1	1	0.99447	0.784	42
68	33.712	2	1	0.21047	0.784	43
69	33.92247	3	0	-1.	0.57353	42
70	34.496	1	1	1.06265	0.784	44
71	35.28	2	1	0.27865	0.784	45
72	35.55865	3	0	-1.	0.50535	44
73	36.064	1	1	0.85383	0.784	46
74	36.848	2	1	0.06983	0.784	47
75	36.91783	3	0	-1.	0.71417	46
76	37.632	1	1	0.717	0.784	48
77	38.349	3	0	-1.	0.067	48
78	38.416	1	1	0.37976	0.784	49
79	38.79576	3	0	-1.	0.40424	49
80	39.2	1	1	0.02985	0.784	50
81	39.22985	3	0	-1.	0.75415	50
82	39.984	1	1	0.36361	0.784	51
83	40.34761	3	0	-1.	0.42039	51
84	40.768	1	1	0.16964	0.784	52
85	40.93764	3	0	-1.	0.61436	52
86	41.552	1	1	0.4989	0.784	53
87	42.0509	3	0	-1.	0.2851	53
88	42.336	1	1	0.09477	0.784	54
89	42.43077	3	0	-1.	0.68923	54
90	43.12	1	1	1.22495	0.784	55
91	43.904	2	1	0.44095	0.784	56
92	44.34495	3	0	-1.	0.34305	55
93	44.688	1	1	4.03252	0.784	57
94	45.472	2	1	3.24852	0.784	58
95	46.256	2	1	2.46452	0.784	59
96	47.04	2	1	1.68052	0.784	60
97	47.824	2	1	0.89652	0.784	61
98	48.608	2	1	0.11252	0.784	62
99	48.72052	3	0	-1.	0.67148	57
100	49.392	1	1	0.14152	0.784	63

Таблица №2

j	$t_3(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
1	0.784	0.05963	0.84363
2	1.568	0.72127	2.28927
3	2.352	2.0171	4.3691
4	3.136	0.	3.136

5	3.92	0.	3.92
6	4.704	0.32372	5.02772
7	5.488	0.4493	5.9373
8	6.272	1.38899	7.66099
9	7.056	0.	7.056
10	7.84	0.29	8.13
11	8.624	2.82485	11.44885
12	9.408	0.	9.408
13	10.192	0.	10.192
14	10.976	0.	10.976
15	11.76	0.00489	11.76489
16	12.544	0.80921	13.35321
17	13.328	0.	13.328
18	14.112	0.53381	14.64581
19	14.896	0.33802	15.23402
20	15.68	0.82055	16.50055
21	16.464	0.	16.464
22	17.248	0.25591	17.50391
23	18.032	1.18827	19.22027
24	18.816	0.	18.816
25	19.6	0.26642	19.86642
26	20.384	1.74246	22.12646
27	21.168	0.	21.168
28	21.952	0.	21.952
29	22.736	0.07253	22.80853
30	23.52	0.21296	23.73296
31	24.304	1.53165	25.83565
32	25.088	0.	25.088
33	25.872	0.00628	25.87828
34	26.656	1.17976	27.83576
35	27.44	0.	27.44
36	28.224	1.00696	29.23096
37	29.008	0.	29.008
38	29.792	1.02886	30.82086
39	30.576	0.	30.576
40	31.36	1.18124	32.54124
41	32.144	0.	32.144
42	32.928	0.99447	33.92247
43	33.712	0.	33.712
44	34.496	1.06265	35.55865
45	35.28	0.	35.28
46	36.064	0.85383	36.91783
47	36.848	0.	36.848

48	37.632	0.717	38.349
49	38.416	0.37976	38.79576
50	39.2	0.02985	39.22985
51	39.984	0.36361	40.34761
52	40.768	0.16964	40.93764
53	41.552	0.4989	42.0509
54	42.336	0.09477	42.43077
55	43.12	1.22495	44.34495
56	43.904	0.	43.904
57	44.688	4.03252	48.72052
58	45.472	0.	45.472
59	46.256	0.	46.256
60	47.04	0.	47.04
61	47.824	0.	47.824
62	48.608	0.	48.608
63	49.392	0.14152	-

Таблица №3

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	37	0.37	18.5739	0.37605
1	63	0.63	30.8181	0.62395
	100	1	49.392	1

Число заявок $J(100)$, поступивших в СМО на интервале $[0, 49.392] = 63$.

Число $JF(100)$ полностью обслуженных заявок на интервале $[0, 49.392] = 37$.

Число $JL(100)$ отклонённых заявок на интервале $[0, 49.392] = 25$.

Доля отклонённых заявок в общем числе поступивших в СМО заявок на интервале $[0, 49.392] = 0.39683$.

Коэффициент простоя прибора на интервале $[0, 49.392] = 0.36304$.

СМО (M|D|1):

Вариант 6; $T_{об} = 0,794$; $\lambda = 1,052$.

Таблица №1

l	$t_{cob}(l)$	Type(l)	состояние СМО C(l) после события l	$t_{ост}(l)$	$t_{ожз}(l)$	j(l)
1	0.40117	1	1	0.794	1.01359	1
2	1.19517	2	0	-1.	0.21959	1
3	1.41476	1	1	0.794	3.16533	2

4	2.20876	2	0	-1.	2.37133	2
5	4.58009	1	1	0.794	0.03231	3
6	4.6124	1	2	0.76169	1.26603	4
7	5.37409	2	1	0.794	0.	3
8	6.16809	2	0	-1.	0.01438	4
9	6.18247	1	1	0.794	0.80838	5
10	6.97647	2	0	-1.	0.01438	5
11	6.99085	1	1	0.794	0.01125	6
12	7.0021	1	2	0.78275	0.69513	7
13	7.69723	1	3	0.08762	0.68404	8
14	7.78485	2	2	0.794	0.	6
15	8.57885	2	1	0.794	0.	7
16	9.37285	2	0	-1.	0.1066	8
17	9.47945	1	1	0.794	0.9006	9
18	10.27345	2	0	-1.	0.1066	9
19	10.38005	1	1	0.794	1.65995	10
20	11.17405	2	0	-1.	0.86595	10
21	12.04	1	1	0.794	1.2449	11
22	12.834	2	0	-1.	0.4509	11
23	13.28489	1	1	0.794	0.17155	12
24	13.45645	1	2	0.62245	0.28961	13
25	13.74606	1	3	0.33284	1.70368	14
26	14.07889	2	2	0.794	0.	12
27	14.87289	2	1	0.794	0.	13
28	15.66689	2	0	-1.	0.25503	14
29	15.92192	1	1	0.794	1.04903	15
30	16.71592	2	0	-1.	0.25503	15
31	16.97094	1	1	0.794	1.56596	16
32	17.76494	2	0	-1.	0.77196	16
33	18.53691	1	1	0.794	0.18997	17
34	18.72687	1	2	0.60403	1.4156	18
35	19.33091	2	1	0.794	0.	17
36	20.12491	2	0	-1.	0.21885	18
37	20.34376	1	1	0.794	1.01285	19
38	21.13776	2	0	-1.	0.21885	19
39	21.35661	1	1	0.794	0.07615	20
40	21.43276	1	2	0.71785	2.87265	21
41	22.15061	2	1	0.794	0.	20
42	25.02326	1	2	0.43047	0.36353	22
43	25.38679	1	3	0.06693	0.92681	23
44	25.45373	2	2	0.794	0.	21
45	26.38053	1	3	0.49097	0.30303	24
46	26.68356	1	4	0.18794	0.08573	25

47	26.7693	1	5	0.1022	0.0623	26
48	26.83159	1	6	0.03991	0.78023	27
49	26.8715	2	5	0.794	0.	22
50	27.6655	2	4	0.794	0.	23
51	28.4595	2	3	0.794	0.	24
52	29.2535	2	2	0.794	0.	25
53	30.0475	2	1	0.794	0.	26
54	30.8415	2	0	-1.	0.73375	27
55	31.57525	1	1	0.794	1.52775	28
56	32.36925	2	0	-1.	0.73375	28
57	33.10301	1	1	0.794	0.12171	29
58	33.22472	1	2	0.67229	0.60564	30
59	33.83036	1	3	0.06665	0.55115	31
60	33.89701	2	2	0.794	0.	29
61	34.69101	2	1	0.794	0.	30
62	35.48501	2	0	-1.	0.8472	31
63	36.3322	1	1	0.794	1.6412	32
64	37.1262	2	0	-1.	0.8472	32
65	37.9734	1	1	0.794	0.61474	33
66	38.58814	1	2	0.17926	0.08273	34
67	38.67087	1	3	0.09653	0.37518	35
68	38.7674	2	2	0.794	0.	33
69	39.14258	1	3	0.78585	0.00815	36
70	39.15073	1	4	0.7777	3.16347	37
71	39.92843	2	3	0.794	0.	34
72	40.72243	2	2	0.794	0.	35
73	41.51643	2	1	0.794	0.	36
74	42.31043	2	0	-1.	0.43667	37
75	42.7471	1	1	0.794	1.23067	38
76	43.5411	2	0	-1.	0.43667	38
77	43.97777	1	1	0.794	0.24757	39
78	44.22533	1	2	0.54643	0.17843	40
79	44.40376	1	3	0.368	0.43756	41
80	44.77177	2	2	0.794	0.	39
81	45.20933	1	3	0.07115	0.72285	42
82	45.28048	2	2	0.794	0.	40
83	46.07448	2	1	0.794	0.	41
84	46.86848	2	0	-1.	0.38935	42
85	47.25783	1	1	0.794	1.18335	43
86	48.05183	2	0	-1.	0.38935	43
87	48.44117	1	1	0.794	1.02804	44
88	49.23517	2	0	-1.	0.23404	44
89	49.46922	1	1	0.794	1.8688	45

90	50.26322	2	0	-1.	1.0748	45
91	51.33802	1	1	0.794	0.68979	46
92	52.02781	1	2	0.10421	0.88749	47
93	52.13202	2	1	0.794	0.	46
94	53.01951	1	2	0.69127	0.10273	48
95	53.12223	1	3	0.58855	0.74444	49
96	53.71078	2	2	0.794	0.	47
97	54.45522	1	3	0.02301	0.77099	50
98	54.47823	2	2	0.794	0.	48
99	55.27223	2	1	0.794	0.	49
100	56.06623	2	0	-1.	0.25905	50

Таблица №2

j	t ₃ (j)	q(j)	t _{оч} (j)	t _{ноб} (j)	t _{обсл} (j)	t _{коб} (j)
1	0.40117	0	0.	0.40117	0.794	1.19517
2	1.41476	0	0.	1.41476	0.794	2.20876
3	4.58009	0	0.	4.58009	0.794	5.37409
4	4.6124	1	0.03231	5.08443	0.794	6.16809
5	6.18247	0	0.	6.18247	0.794	6.97647
6	6.99085	0	0.	6.99085	0.794	7.78485
7	7.0021	1	0.01125	7.58727	0.794	8.57885
8	7.69723	2	1.50038	6.99085	0.794	9.37285
9	9.47945	0	0.	9.47945	0.794	10.27345

10	10.38005	0	0.	10.38005	0.794	11.17405
11	12.04	0	0.	12.04	0.794	12.834
12	13.28489	0	0.	13.28489	0.794	14.07889
13	13.45645	1	0.17155	14.65574	0.794	14.87289
14	13.74606	2	1.25516	13.28489	0.794	15.66689
15	15.92192	0	0.	15.92192	0.794	16.71592
16	16.97094	0	0.	16.97094	0.794	17.76494
17	18.53691	0	0.	18.53691	0.794	19.33091
18	18.72687	1	0.18997	19.34847	0.794	20.12491
19	20.34376	0	0.	20.34376	0.794	21.13776
20	21.35661	0	0.	21.35661	0.794	22.15061
21	21.43276	1	0.07615	23.51141	0.794	25.45373
22	25.02326	1	0.36353	25.5196	0.794	26.8715
23	25.38679	2	1.52107	26.81782	0.794	27.6655
24	26.38053	2	1.09703	26.0775	0.794	28.4595

25	26.68356	3	2.19406	26.8715	0.794	29.2535
26	26.7693	4	3.0738	27.6655	0.794	30.0475
27	26.83159	5	3.93009	28.4595	0.794	30.8415
28	31.57525	0	0.	31.57525	0.794	32.36925
29	33.10301	0	0.	33.10301	0.794	33.89701
30	33.22472	1	0.12171	33.58751	0.794	34.69101
31	33.83036	2	1.52135	33.10301	0.794	35.48501
32	36.3322	0	0.	36.3322	0.794	37.1262
33	37.9734	0	0.	37.9734	0.794	38.7674
34	38.58814	1	0.61474	38.25205	0.794	39.92843
35	38.67087	2	1.49147	41.5202	0.794	40.72243
36	39.14258	2	0.80215	39.13443	0.794	41.51643
37	39.15073	3	1.6043	39.92843	0.794	42.31043
38	42.7471	0	0.	42.7471	0.794	43.5411
39	43.97777	0	0.	43.97777	0.794	44.77177

40	44.22533	1	0.24757	44.04733	0.794	45.28048
41	44.40376	2	1.22	45.13818	0.794	46.07448
42	45.20933	2	1.51685	44.48648	0.794	46.86848
43	47.25783	0	0.	47.25783	0.794	48.05183
44	48.44117	0	0.	48.44117	0.794	49.23517
45	49.46922	0	0.	49.46922	0.794	50.26322
46	51.33802	0	0.	51.33802	0.794	52.13202
47	52.02781	1	0.68979	52.12129	0.794	53.71078
48	53.01951	1	0.10273	53.07267	0.794	54.47823
49	53.12223	2	0.99945	54.43221	0.794	55.27223
50	54.45522	2	1.56499	53.68423	0.794	56.06623

Таблица №3

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	24	0.24	12.39339	0.22105
1	35	0.35	24.97117	0.44539
2	22	0.22	12.54758	0.2238
3	13	0.13	3.60046	0.06422
4	3	0.03	1.65743	0.02956
5	2	0.02	0.8563	0.01527
6	1	0.01	0.03991	0.00071
	100	1	56.06623	1

Число заявок $J(100)$, поступивших в СМО на интервале $[0, 56.06623]$ = 50.

Число $JF(100)$ полностью обслуженных заявок на интервале $[0, 56.06623]$ = 50.

Среднее число заявок, находившихся в СМО, на интервале $[0, 56.06623]$ = 1.45.

Среднее время пребывания заявок в очереди на интервале $[0, 56.06623]$ = 0.55827.

Среднее время пребывания заявок в СМО на интервале $[0, 56.06623]$ = 1.44037.

Коэффициент простоя прибора на интервале $[0, 56.06623]$ = 0.21851.

СМО (M|M|1):

Вариант 6. $\lambda = 1,052$. $\mu = 1,254$.

Таблица №1

l	$t_{\text{cob}}(l)$	Type(l)	состояние СМО C(l) после события l	$t_{\text{ост}}(l)$	$t_{\text{ожз}}(l)$	j(l)
1	2.20718	1	1	2.93613	0.2653	1
2	2.47248	1	2	2.67083	0.10326	2
3	2.57574	1	3	2.56757	0.12289	3
4	2.69863	1	4	2.44469	2.39239	4
5	5.09102	1	5	0.0523	0.22729	5
6	5.14332	2	4	1.56938	0.	1
7	5.37061	1	5	0.8503	0.71909	6
8	6.0897	1	6	0.13121	0.19363	7
9	6.22091	2	5	1.52942	0.	2
10	6.41454	1	6	0.21715	1.31226	8
11	6.63169	2	5	0.08391	0.	3
12	6.71561	2	4	0.59168	0.	4
13	7.30728	2	3	0.03554	0.	5
14	7.34283	2	2	1.44913	0.	6
15	8.65509	1	3	0.85524	0.59389	9
16	9.24898	1	4	0.26135	0.24182	10
17	9.4908	1	5	0.01953	0.02465	11
18	9.51033	2	4	0.55305	0.	7
19	9.53498	1	5	0.10028	0.45278	12
20	9.63526	2	4	3.01199	0.	8
21	10.08803	1	5	2.40352	0.60848	13
22	10.69651	1	6	1.79504	0.02775	14

23	10.72426	1	7	1.76729	1.13163	15
24	11.8559	1	8	0.63565	1.58445	16
25	12.49155	2	7	2.3509	0.	9
26	14.84244	2	6	0.38485	0.	10
27	15.2273	2	5	0.50454	0.	11
28	15.73183	2	4	0.94611	0.	12
29	16.67794	2	3	1.26553	0.	13
30	17.94348	2	2	0.49364	0.	14
31	18.43712	2	1	0.39024	0.	15
32	18.82736	2	0	-1.	2.64338	16
33	21.47074	1	1	3.37338	3.03362	17
34	24.50436	1	2	0.33976	0.43994	18
35	24.84412	2	1	3.11338	0.	17
36	25.28406	1	2	2.88099	0.23239	19
37	25.51644	1	3	2.6486	0.43077	20
38	25.94722	1	4	2.21783	0.46032	21
39	26.40754	1	5	1.75751	0.08829	22
40	26.49583	1	6	1.66922	0.16521	23
41	26.66103	1	7	1.50401	0.31596	24
42	26.977	1	8	1.18805	1.37002	25
43	28.16505	2	7	0.88519	0.	18
44	29.53506	1	8	0.32019	0.56499	26
45	29.85526	2	7	0.05873	0.	19
46	29.91399	2	6	0.41225	0.	20
47	30.32624	2	5	0.22171	0.	21
48	30.54795	2	4	1.12859	0.	22
49	31.11294	1	5	0.69538	0.4332	27
50	31.54614	1	6	0.26218	0.3878	28
51	31.80832	2	5	0.09027	0.	23
52	31.89859	2	4	0.14492	0.	24
53	32.04351	2	3	0.05658	0.	25
54	32.10009	2	2	0.71193	0.	26
55	32.48789	1	3	0.29774	0.41419	29
56	32.78563	2	2	0.90408	0.	27
57	33.68971	2	1	4.02732	0.	28
58	34.1039	1	2	0.93669	3.09063	30
59	35.04059	2	1	0.43715	0.	29
60	35.47774	2	0	-1.	0.56685	30
61	36.04459	1	1	0.16095	1.004	31
62	36.20554	2	0	-1.	0.84305	31
63	37.04858	1	1	0.2612	1.16567	32
64	37.30979	2	0	-1.	0.90447	32
65	38.21426	1	1	0.03039	1.64222	33

66	38.24465	2	0	-1.	1.61183	33
67	39.85648	1	1	0.5647	1.05018	34
68	40.42118	2	0	-1.	0.48548	34
69	40.90666	1	1	1.00129	0.17518	35
70	41.08184	1	2	0.8261	2.14681	36
71	41.90794	2	1	0.27255	0.	35
72	44.05476	1	2	0.035	0.23756	37
73	44.08975	2	1	1.28102	0.	36
74	45.37077	2	0	-1.	1.27916	37
75	46.64994	1	1	0.17775	2.56018	38
76	46.82769	2	0	-1.	2.38243	38
77	49.21012	1	1	0.07506	0.15163	39
78	49.28518	2	0	-1.	0.07657	39
79	49.36175	1	1	0.11575	0.31403	40
80	49.4775	2	0	-1.	0.19828	40
81	49.67578	1	1	1.57348	0.19054	41
82	49.86632	1	2	1.38294	0.11435	42
83	49.98067	1	3	1.26859	0.533	43
84	50.51367	1	4	0.73559	2.18066	44
85	51.24926	2	3	0.34434	0.	41
86	51.5936	2	2	0.83776	0.	42
87	53.77426	1	3	0.02858	0.80918	45
88	53.80284	2	2	0.23532	0.	43
89	54.03816	2	1	0.13243	0.	44
90	54.17059	2	0	-1.	0.46775	45
91	54.63834	1	1	0.35127	0.60018	46
92	54.98961	2	0	-1.	0.2489	46
93	55.23852	1	1	0.98344	1.70391	47
94	56.22195	2	0	-1.	0.72048	47
95	56.94243	1	1	0.28769	4.07399	48
96	57.23012	2	0	-1.	3.7863	48
97	61.01642	1	1	1.69061	3.3069	49
98	62.70703	2	0	-1.	1.61629	49
99	64.32332	1	1	2.12868	1.02838	50
100	65.3517	1	2	1.10029	0.31784	51

Таблица №2

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
1	2.20718	0	0	2.20718	2.93613	5.14332
2	2.47248	1	-1.10145	2.38218	1.56938	6.22091

3	2.57574	2	0.53123	4.71395	1.52942	6.63169
4	2.69863	3	0.73803	6.19739	0.08391	6.71561
5	5.09102	4	3.72209	6.54778	0.59168	7.30728
6	5.37061	4	1.39025	6.12393	0.03554	7.34283
7	6.0897	5	3.55847	7.27174	1.44913	9.51033
8	6.41454	5	2.49616	8.06632	0.55305	9.63526
9	8.65509	2	2.70981	9.4347	3.01199	12.49155
10	9.24898	3	5.65459	10.42835	2.3509	14.84244
11	9.4908	4	6.28126	10.14065	0.38485	15.2273
12	9.53498	4	6.152	14.45759	0.50454	15.73183
13	10.08803	4	1.78288	14.72276	0.94611	16.67794
14	10.69651	5	3.65689	14.78572	1.26553	17.94348
15	10.72426	6	4.17828	15.41241	0.49364	18.43712
16	11.8559	7	5.70016	17.44984	0.39024	18.82736
17	21.47074	0	0.	21.47074	3.37338	24.84412
18	24.50436	1	2.77362	21.57092	3.11338	28.16505
19	25.28406	1	-1.9958	25.23364	0.88519	29.85526
20	25.51644	2	-1.70469	29.21487	0.05873	29.91399
21	25.94722	3	-0.86166	29.79653	0.41225	30.32624
22	26.40754	4	-0.17963	29.50174	0.22171	30.54795
23	26.49583	5	1.03724	30.10453	1.12859	31.80832
24	26.66103	6	1.29272	30.80536	0.09027	31.89859
25	26.977	7	1.7536	31.71805	0.14492	32.04351
26	29.53506	7	1.79285	31.75367	0.05658	32.10009
27	31.11294	4	0.30832	31.98692	0.71193	32.78563
28	31.54614	5	1.6456	32.19014	0.90408	33.68971
29	32.48789	2	4.63366	31.88155	4.02732	35.04059
30	34.1039	1	-0.49955	33.1672	0.43715	35.47774
31	36.04459	0	0.	36.04459	0.16095	36.20554
32	37.04858	0	0.	37.04858	0.2612	37.30979
33	38.21426	0	0.	38.21426	0.03039	38.24465
34	39.85648	0	0.	39.85648	0.5647	40.42118
35	40.90666	0	0.	40.90666	1.00129	41.90794
36	41.08184	1	-0.55355	42.22737	0.27255	44.08975
37	44.05476	1	1.24602	44.01976	1.28102	45.37077
38	46.64994	0	0.	46.64994	0.17775	46.82769
39	49.21012	0	0.	49.21012	0.07506	49.28518
40	49.36175	0	0.	49.36175	0.11575	49.4775
41	49.67578	0	0.	49.67578	1.57348	51.24926
42	49.86632	1	-1.0386	51.12085	0.34434	51.5936
43	49.98067	2	-0.08648	50.90492	0.83776	53.80284
44	50.51367	3	0.68183	53.74568	0.23532	54.03816
45	53.77426	2	0.33917	53.56753	0.13243	54.17059

46	54.63834	0	0.	54.63834	0.35127	54.98961
47	55.23852	0	0.	55.23852	0.98344	56.22195
48	56.94243	0	0.	56.94243	0.28769	57.23012
49	61.01642	0	0.	61.01642	1.69061	62.70703
50	64.32332	0	0.	64.32332	2.93613	5.14332
51	65.3517	1	0.64294	2.20718	1.56938	6.22091

Таблица 3

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	15	0.15	20.03841	0.30662
1	23	0.23	14.63361	0.22392
2	14	0.14	8.10131	0.12396
3	10	0.1	3.70888	0.05675
4	11	0.11	6.78254	0.10379
5	12	0.12	3.11523	0.04767
6	7	0.07	1.6006	0.02449
7	5	0.05	5.22724	0.07999
8	3	0.03	2.14389	0.03281
	100	1	65.3517	1

Число заявок $J(100)$, поступивших в СМО на интервале $[0, 65.3517] = 51$.

Число $JF(100)$ полностью обслуженных заявок на интервале $[0, 65.3517] = 49$.

Среднее число заявок, находившихся в СМО, на интервале $[0, 65.3517] = 2.85$.

Среднее время пребывания заявок в очереди на интервале $[0, 65.3517] = 1.19752$.

Среднее время пребывания заявок в СМО на интервале $[0, 65.3517] = 0.35329$.

Коэффициент простоя прибора на интервале $[0, 65.3517] = 0.27285$.

Список литературы

1. Кирпичников А.П. Методы прикладной теории массового обслуживания. – М.: URSS, 2018 – 224 с.
2. Ивченко Г.И., Каштанов В.А., Коваленко И.Н. Теория массового обслуживания. – М.: URSS, 2012 – 304 с
3. Введение в теорию массового обслуживания [Электронный ресурс]: учебное пособие для студентов, обучающихся по направлению «Информационные системы и технологии» / Е. К. Белый. --- Петрозаводск: Издательство ПетрГУ, 2014 – 76 с.
4. Лобузов А.А., Гумляева С.Д., Норин Н.В. Задачи по теории случайных процессов. – М.: МИРЭА, 1993 – 68 с.

Приложение

```
import numpy as np
from scipy.stats import expon
from decimal import Decimal

f = open('answer.txt', 'r+')

u = 1.254
lambda = 1.052
Ts = 0.794
Tt = 0.784 # Время между заявками

L = list(range(1, 101))
Ttime = []
Ttype = []
condition = []
Tremained = []
Tnew = []
numj = []

jN = np.zeros(100)
jP = np.zeros(100)
jS = np.zeros(100)
jF = np.zeros(100)

T0 = Tt
T1 = 0

J5 = 0
JF5 = 0
JL5 = 0
Dl = 0
Tl = 0

service_time = expon.rvs(scale=1 / u, size=100)
service_time = [0.0596276, 0.72127201, 2.01710223, 0.32372014,
0.44929662, 1.38898635, 0.28999866, 2.82485371,
0.00489021, 0.80920563, 0.53380782, 0.33801901, 0.82054915,
0.25590868, 1.18827375, 0.26642167, 1.7424584,
0.07252952, 0.212955, 1.53165004, 0.00627624, 1.17975602,
1.00696473, 1.02885748, 1.1812449, 0.99447012,
```



```

1.0626488, 0.85383117, 0.71700256, 0.3797618, 0.02984504,
0.36360791, 0.16964442, 0.49890133, 0.0947652,
1.22494775, 4.03252267, 0.14152315, 0.50042083, 0.39265258,
1.88307502, 0.2069792, 0.85826512, 0.80996065,
1.12609482, 1.31039982, 1.51896547, 0.85907068, 1.14635457,
1.60915022, 0.17615997, 0.15130618, 0.15950662,
0.0796058, 0.35177219, 0.18433293, 0.6734803, 1.12783108,
0.04402474, 1.18628942, 0.02226666, 0.1077807, 1.06877713,
0.0984029, 0.27811944, 0.51479237, 0.38918731, 0.13941547,
0.19589416, 0.35329383, 0.0173077, 1.53808859,
0.023749, 0.10802429, 0.09758635, 0.38637948, 1.23661508,
1.059142, 0.14084473, 0.01654323, 1.26470771, 0.28802272,
0.03494353, 0.33704994, 0.16819181, 0.5593081, 0.29836402,
0.46637185, 0.97799658, 0.44816637, 0.16128861, 1.01136497,
2.1438003, 0.16640893, 0.15373774, 0.02907984, 1.05664218,
0.32663872, 1.32075165, 1.1589173]

```

```

print(service_time)

```

```

i = 0

```

```

j = 0

```

```

jN[j] = j + 1

```

```

jP[j] = Tt

```

```

jS[j] = service_time[i]

```

```

Ttime.append(Tt)

```

```

Ttype.append(1)

```

```

condition.append(1)

```

```

T1 += service_time[i]

```

```

Tremained.append(service_time[i])

```

```

Tnew.append(Tt)

```

```

j += 1

```

```

jr = j

```

```

numj.append(j)

```

```

trimen = service_time[i] - Tt

```

```

S_con = 1

```

```

J5+=1

```

```

while len(Ttime) != 100:

```

```

    if S_con == 0:

```

```

        T0 += -trimen

```

```

jN[j] = j + 1
jP[j] = Ttime[-1] - trimen
jS[j] = service_time[i]

Time.append(Ttime[-1] - trimen)
Ttype.append(1)
condition.append(1)

T1 += service_time[i]

Tremained.append(service_time[i])
Tnew.append(Tt)
j += 1
jr = j
numj.append(j)
trimen = service_time[i] - Tt
S_con = 1
J5 += 1
elif trimen < 0 and Ttype[-1] == 2:

    jF[jr-1] = Ttime[-1] + Tremained[-1]

    Time.append(Ttime[-1] + Tremained[-1])
    Ttype.append(3)
    condition.append(0)
    Tremained.append(-1)

    T0 += -trimen

    Tnew.append(-trimen)
    Tl += -trimen
    numj.append(jr)
    i += 1
    S_con = 0
    JF5 += 1
elif S_con == 1 and trimen > 0:

    jN[j] = j + 1
    jP[j] = Ttime[-1] + Tt
    jS[j] = 0
    jF[j] = Ttime[-1] + Tt

    Time.append(Ttime[-1] + Tt)
    Ttype.append(2)

```

```

condition.append(1)
Tremained.append(trimen)
Tnew.append(Tt)
j += 1
numj.append(j)
trimen -= Tt
J5 += 1
JL5 += 1
elif trimen < 0:

    jF[jr-1] = Ttime[-1] + Tremained[-1]

    Ttime.append(Ttime[-1] + Tremained[-1])
    Ttype.append(3)
    condition.append(0)
    Tremained.append(-1)

    T0 += -trimen

    Tnew.append(-trimen)
    numj.append(jr)
    i += 1
    S_con = 0
    Tl += -trimen
    JF5 += 1

R0 = condition.count(0)
R1 = condition.count(1)
v0 = R0 / 100
v1 = R1 / 100
T0 = Ttime[-1] - Tl
delta0 = T0 / Ttime[-1]
delta1 = Tl / Ttime[-1]
Dl = JL5/J5
Tl = Tl/Ttime[-1]

f.write(str(np.around(L, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttime, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttype, 5)))
f.write('\n')

```

```

f.write('\n')
f.write('\n')
f.write(str(np.around(condition, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tremained, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tnew, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(numj, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(jN, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jP, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jS, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jF, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(R0, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(R1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(v0, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(v1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(T0, 5)))
f.write('\n')

```

```

f.write('\n')
f.write(str(np.around(T1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(delta0, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(delta1, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(J5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(JF5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(JL5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(D1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(T1, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')

```

```

Ttime = []
Ttype = []
condition = []
Tremained = []
Tnew = []
numj = []

```

```

jN = np.zeros(100)
jP = np.zeros(100)
jQ = np.zeros(100)

```

```

jQt = np.zeros(100)
jS = np.zeros(100)
jD = np.zeros(100)
jF = np.zeros(100)

R = np.zeros(100)
V = np.zeros(100)

T0 = Tt
T1 = 0

arrive_time = expon.rvs(scale=1 / lambd, size=100)
arrive_time = [0.40117091, 1.0135882, 3.16533116, 0.03230948,
1.26603334, 0.80838107, 0.01125001, 0.69512731, 0.68403592, 0.90059743,
1.65994806, 1.24489832, 0.17155369, 0.28960933, 1.70368105, 1.04902514,
1.56596271, 0.18996548, 1.41560046, 1.01285334, 0.07614671, 2.87264663,
0.3635338, 0.9268059, 0.30303236, 0.0857338, 0.06229516, 0.7802278,
1.52775425, 0.12170897, 0.60564453, 0.55114783, 1.64119539, 0.6147427,
0.08273187, 0.37517914, 0.00815103, 3.16347353, 1.23067033, 0.24756659,
0.17842878, 0.43756393, 0.72285051, 1.18334614, 1.02804347, 1.86880286,
0.68978814, 0.88748695, 0.10272516, 0.74443909, 0.7709879, 1.05304843,
0.74895205, 0.88201039, 1.55375056, 0.02729069, 0.89429496, 0.65826222,
0.25342226, 0.25016314, 1.14511598, 1.14859708, 0.94699604, 0.63564956,
0.44301601, 0.18424226, 0.48948496, 0.21349709, 1.06094758, 0.14500643,
0.99892129, 3.42469012, 0.85917016, 2.33556292, 0.72154634, 0.18810993,
0.36188151, 0.13434438, 1.99659399, 1.41253129, 1.26997202, 0.28451984,
0.93023139, 0.62328968, 0.89739204, 1.07336679, 0.03165158, 1.46436741,
0.48715263, 1.61223369, 0.2703513, 0.34097123, 0.38275818, 1.89983983,
2.47692569, 0.60724232, 0.83825833, 1.27701367, 0.37523242, 0.11070942]
print(arrive_time)
i = 0
it = i

S_con = 0
trimen = 0

R[0] += 1
V[0] += arrive_time[i]

J5 = 0
JF5 = 0
Z5 = 0
Tl = 0

```

```

Ttime.append(arrive_time[i])

jN[it] = i + 1
jP[it] = arrive_time[i]
jQ[it] = 0
jQt[it] = 0
jS[it] = arrive_time[i]

Ttype.append(1)
condition.append(1)

# T1 += service_time[i]

Tremained.append(Ts)
Tnew.append(arrive_time[i + 1])
numj.append(it + 1)
trimen = arrive_time[it + 1] - Ts
S_con = 1
R[S_con] += 1
#V[S_con] += min(Ts, arrive_time[i + 1])
i+=1
it+=1
J5 += 1
while len(Ttime) != 100:
    if S_con == 0:

        jN[i] = i + 1
        jP[i] = Ttime[-1] + Tnew[-1]
        jQ[i] = 0
        jQt[i] = 0
        jS[i] = Ttime[-1] + Tnew[-1]

        V[S_con] += Tnew[-1]
        Ttime.append(Ttime[-1] + Tnew[-1])
        Ttype.append(1)
        condition.append(S_con + 1)

        # T1 += service_time[i]

        Tremained.append(Ts)
        Tnew.append(arrive_time[it + 1])
        numj.append(it + 1)
        trimen = arrive_time[it + 1] - Ts
        S_con += 1

```

```

R[S_con] += 1
#V[S_con] += min(Ts, arrive_time[it + 1])
it += 1
i += 1
J5 += 1
Z5 += S_con
elif trimen > 0:

    jS[it] = Ttime[-1] + Tnew[-1] - Ts
    jD[it] = Ts
    jF[it] = Ttime[-1] + Tremained[-1]

    #V[S_con] += Tremained[-1]
    Ttime.append(Ttime[-1] + Tremained[-1])
    Ttype.append(2)
    V[S_con] += Tremained[-1]
    condition.append(S_con - 1)
    if S_con - 1 == 0:
        Tremained.append(-1)
        Tnew.append(trimen)
        #V[S_con-1] += trimen
        Tl += trimen
    else:
        #V[S_con-1] += Tremained[-1]
        Tremained.append(Ts)
        Tnew.append(0)
        #V[S_con-1] += min(Ts, trimen)

    # T0 += -trimen

    trimen = arrive_time[i + 1] - Ts
    numj.append(it)
    if S_con - 1 != 0:
        it += 1
        S_con -= 1
        R[S_con] += 1
        JF5 += 1
        Z5 += S_con
    elif trimen < 0:

        jN[i] = i + 1
        jP[i] = Ttime[-1] + arrive_time[i]
        jQ[i] = S_con
        jQt[i] = Ts * (S_con) + trimen

```



```

V[S_con] += arrive_time[i]
Time.append(Ttime[-1] + arrive_time[i])
Ttype.append(1)
condition.append(S_con + 1)
Tremained.append(-trimen)
Tnew.append(arrive_time[i + 1])
i += 1
S_con += 1
R[S_con] += 1
#V[S_con] += min(-trimen, arrive_time[i])
numj.append(i)
trimen += arrive_time[i]
J5 += 1
Z5 += S_con

```

```

Rot = R/100
Vot = V/Ttime[-1]
print(sum(V))
print(Ttime[-1])
Z5 = Z5/100
Tq5 = sum(jQt)/JF5
Tm5 = sum(jF-jP)/JF5
Tl = Tl/Ttime[-1]

```

```

f.write(str(np.around(L, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttime, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttype, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(condition, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tremained, 5)))
f.write('\n')

```

```

f.write('\n')
f.write(str(np.around(Tnew, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(numj, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(jN, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jP, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jQ, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jQt, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jS, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jD, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jF, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(R, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(V, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Rot, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Vot, 5)))
f.write('\n')
f.write('\n')

```

```

f.write(str(np.around(J5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(JF5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Z5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tq5, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(Tm5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tl, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
print(sum(V))

```

```

Ttime = []
Ttype = []
condition = []
Tremained = []
Tnew = []
numj = []

```

```

jN = np.zeros(100)
jP = np.zeros(100)
jQ = np.zeros(100)
jQt = np.zeros(100)
jS = np.zeros(100)

```

```
jD = np.zeros(100)
jF = np.zeros(100)
```

```
R = np.zeros(100)
V = np.zeros(100)
```

```
T0 = Tt
T1 = 0
```

```
arrive_time = expon.rvs(scale=1 / lambda, size=105)
arrive_time = [2.20718492, 0.26529921, 0.10325718, 0.12288507,
2.39239022, 0.22729359, 0.71908768, 0.19363402, 1.31226468, 0.59389162,
0.24181767, 0.02464913, 0.45277553, 0.60847666, 0.02775334, 1.13163481,
1.5844468, 3.03362209, 0.43993822, 0.23238649, 0.43077442, 0.4603215,
0.08828852, 0.16520653, 0.31596391, 1.37001638, 0.56499403, 0.43320493,
0.38779847, 0.41418553, 3.09062796, 1.00399573, 1.16567395, 1.6422182,
1.05017879, 0.17518421, 2.14681417, 0.23755632, 2.56018167, 0.15162922,
0.3140315, 0.19054134, 0.11435043, 0.53299786, 2.18066243, 0.80918078,
0.60017746, 1.70391298, 4.07399, 3.30689993, 1.02838463, 0.31783687,
1.03291466, 0.11887342, 0.14481511, 0.07579462, 0.63721745, 1.95042728,
0.15373085, 2.49893149, 1.51821568, 2.30314229, 0.31068136, 1.45814779,
0.80388062, 0.60584391, 0.89974687, 0.06774533, 0.5994367, 0.15957083,
0.82415271, 0.35558941, 0.54138821, 3.28102585, 0.23260546, 2.12848419,
0.99654792, 0.33818901, 0.11495295, 0.04273647, 0.03962951, 0.17755221,
0.61092236, 2.19789821, 1.02850586, 0.44404132, 0.80435907, 1.84632329,
2.56700775, 2.23089819, 0.2723861, 0.97549922, 0.34930668, 0.18054849,
0.69297077, 0.79819385, 4.04333174, 0.3659875, 0.91265219, 2.01531124,
0.05463354, 0.45549287, 1.52315717, 1.2002214, 0.7876374,]
print(arrive_time)
service_time = expon.rvs(scale=1 / u, size=105)
service_time = [2.93613062e+00, 1.56938416e+00, 1.52941779e+00,
8.39128275e-02, 5.91677518e-01, 3.55420398e-02, 1.44913065e+00,
5.53054285e-01, 3.01199278e+00, 2.35089535e+00, 3.84852334e-01,
5.04537144e-01, 9.46109388e-01, 1.26553492e+00, 4.93639929e-01,
3.90237738e-01, 3.37337979e+00, 3.11337732e+00, 8.85186315e-01,
5.87309448e-02, 4.12249674e-01, 2.21708319e-01, 1.12858638e+00,
9.02687314e-02, 1.44916871e-01, 5.65828381e-02, 7.11930058e-01,
9.04080382e-01, 4.02732241e+00, 4.37148373e-01, 1.60947711e-01,
2.61202992e-01, 3.03890393e-02, 5.64698266e-01, 1.00128821e+00,
2.72552891e-01, 1.28101678e+00, 1.77751875e-01, 7.50633058e-02,
1.15748907e-01, 1.57348018e+00, 3.44341645e-01, 8.37761906e-01,
2.35317669e-01, 1.32429834e-01, 3.51273978e-01, 9.83436506e-01,
2.87693584e-01, 1.69060553e+00, 2.12867780e+00, 1.74323242e+00,
9.39580009e-01, 2.01175799e-01, 5.71501694e-02, 4.47566636e-01,
```

```

2.21465944e+00, 1.25437703e+00, 2.82898191e-01, 1.55794780e+00,
5.25302804e-01, 1.49683128e+00, 8.78322611e-01, 6.94661885e-02,
1.30313830e-01, 3.02726805e-01, 3.19998528e+00, 4.81134657e-01,
7.21457649e-01, 4.93482834e-02, 4.36012464e-01, 6.39453990e-01,
8.99441309e-02, 2.45257555e+00, 1.70642016e+00, 6.14330704e-01,
9.13332468e-01, 2.26219077e-01, 1.93888673e-01, 1.30890614e+00,
1.38956501e-03, 5.43608367e-01, 6.39416687e-02, 9.02842252e-01,
2.11538645e+00, 3.80504429e-02, 2.69041841e-01, 1.20720293e+00,
5.36911339e-01, 1.17333053e+00, 1.34255053e-01, 8.01150684e-01,
8.24543871e-01, 8.39682266e-01, 1.32277362e+00, 1.95569786e-01,
1.08849715e-01, 2.01899368e-01, 1.73524631e+00, 3.05655782e-01,
2.76517344e-01, 5.15472337e-01, 9.14255974e-01, 1.54729761e+00,
2.49196005e-01, 1.50307918e+00,]

```

```

    print(service_time)

```

```

    i = 0

```

```

    j = 0

```

```

    it = i

```

```

    S_con = 0

```

```

    trimen = 0

```

```

    R[0] += 1

```

```

    V[0] += arrive_time[i]

```

```

    J5 = 0

```

```

    JF5 = 0

```

```

    Z5 = 0

```

```

    Tl = 0

```

```

    Ttime.append(arrive_time[i])

```

```

    jN[it] = i + 1

```

```

    jP[it] = arrive_time[i]

```

```

    jQ[it] = 0

```

```

    jQt[it] = 0

```

```

    jS[it] = arrive_time[i]

```

```

    Ttype.append(1)

```

```

    condition.append(1)

```

```

    # Tl += service_time[i]

```

```

    Tremained.append(service_time[j])

```

```

    Tnew.append(arrive_time[i + 1])

```

```

numj.append(it + 1)
trimen = arrive_time[it + 1] - service_time[j]
S_con = 1
R[S_con] += 1
#V[S_con] += min(service_time[j], arrive_time[i + 1])
i+=1
it+=1
J5 += 1
while len(Ttime) != 100:
    if S_con == 0:

        jN[i] = i + 1
        jP[i] = Ttime[-1] + Tnew[-1]
        jQ[i] = 0
        jQt[i] = 0
        jS[i] = Ttime[-1] + Tnew[-1]

        V[S_con] += Tnew[-1]
        Ttime.append(Ttime[-1] + Tnew[-1])
        Ttype.append(1)
        condition.append(S_con + 1)

        # T1 += service_time[i]

        Tremained.append(service_time[j])
        Tnew.append(arrive_time[it + 1])
        numj.append(it + 1)
        trimen = arrive_time[it + 1] - service_time[j]
        S_con += 1
        R[S_con] += 1
        #V[S_con] += min(service_time[j], arrive_time[i + 1])
        it += 1
        i += 1
        J5 += 1
        Z5 += S_con
    elif trimen > 0:

        jS[it] = Ttime[-1] + Tnew[-1] - service_time[j]
        jD[it] = service_time[j]
        jF[it] = Ttime[-1] + Tremained[-1]

        Ttime.append(Ttime[-1] + Tremained[-1])
        Ttype.append(2)
        V[S_con] += Tremained[-1]

```

```

condition.append(S_con - 1)
j += 1
if S_con - 1 == 0:
    Tremained.append(-1)
    Tnew.append(trimen)
    #V[S_con-1] += trimen
    Tl += trimen
else:
    #V[S_con - 1] += Tremained[-1]
    Tremained.append(service_time[j])
    Tnew.append(0)
    #V[S_con-1] += min(service_time[j], trimen)

# T0 += -trimen

trimen = arrive_time[i + 1] - service_time[j]
numj.append(it)
if S_con - 1 != 0:
    it += 1
    S_con -= 1
    R[S_con] += 1
    JF5 += 1
    Z5 += S_con
elif trimen < 0:

    V[S_con] += arrive_time[i]
    jN[i] = i + 1
    jP[i] = Ttime[-1] + arrive_time[i]
    jQ[i] = S_con
    for k in range(S_con):
        jQt[i] += service_time[j+k+1]
    jQt[i] += trimen

    Ttime.append(Ttime[-1] + arrive_time[i])
    Ttype.append(1)
    condition.append(S_con + 1)
    Tremained.append(-trimen)
    Tnew.append(arrive_time[i + 1])
    i += 1
    S_con += 1
    R[S_con] += 1
    #V[S_con] += min(-trimen, arrive_time[i])
    numj.append(i)
    trimen += arrive_time[i]

```

```

J5 += 1
Z5 += S_con

```

```

Rot = R/100
print(sum(V))
print(Ttime[-1])
Vot = V/Ttime[-1]
Z5 = Z5/100
Tq5 = sum(jQt)/JF5
Tm5 = sum(jF-jP)/JF5
Tl = Tl/Ttime[-1]

```

```

f.write(str(np.around(L, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttime, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Ttype, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(condition, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tremained, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tnew, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(numj, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(jN, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jP, 5)))
f.write('\n')

```



```

f.write('\n')
f.write(str(np.around(jQ, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jQt, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jS, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jD, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jF, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(R, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(V, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Rot, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Vot, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(J5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(JF5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Z5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Tq5, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')

```

```
f.write(str(np.around(Tm5, 5)))  
f.write('\n')  
f.write('\n')  
f.write(str(np.around(Tl, 5)))  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
f.write('\n')  
print(sum(V))
```