



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 14

Тема: Одноканальные системы массового обслуживания

Выполнил:
Студент 4-го курса
Малов И.М.

Группа: КМБО-01-20

МОСКВА 2023

Оглавление

<u>Оглавление</u>	2
<u>Задание</u>	3
<u>Теоретические сведения</u>	8
<u>Средства языка программирования</u>	11
<u>Результаты расчетов</u>	12
<u>Задание 1</u>	12
<u>Задание 2</u>	17
<u>Задание 3</u>	22
<u>Список литературы</u>	27
<u>Приложение</u>	28

Задание

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени 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 (пример 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.

I. 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 равен $t_{\text{собр}}(l) = \Delta T_3$. 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 заданным параметром μ .

II. Одноканальная СМО с бесконечной очередью ($M/D/1/0$).

Дано:

- среднее число заявок λ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром λ);
- время обслуживания заявки прибором $T_{\text{об}}$ (заданная постоянная величина)

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

III. Одноканальная СМО с бесконечной очередью ($M/M/1/0$).

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 $t = 0$ СМО 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 $t_{\text{обсл}}$ в соответствии с показательным законом распределения с параметром μ .

Требуется:

Провести моделирование первых 100 событий в развитии каждой СМО.

1) Составить таблицу 1 с данными о событиях:

- Номер события ;
- Момент наступления события $t_{\text{соб}}(l)$;;
- Тип события $\text{Type}(l)$
- Состояние СМО $C(l)$ после события l ;
- Оставшееся время $t_{\text{ост}}(l)$ обслуживания прибором заявки после события l (если после события прибор свободен, то $t_{\text{ост}}(l) = -1$);
- Время ожидания $t_{\text{ожз}}(l)$, через которое после события в СМО появится новая заявка;
- Номер заявки $j(l)$, участвующей в событии .

2) Для СМО (D|M|1|0) составить таблицу 2 с данными о всех поступивших заявках:

- номер заявки j ;
- момент $t_3(j)$ появления заявки j в СМО;
- время $t_{\text{обсл}}(j)$ обслуживания прибором заявки j ; — момент $t_{\text{коб}}(j)$ 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 $t_{\text{обсл}}(j) = 0$ и $t_{\text{коб}}(j) = t_3(j)$. 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 $t_{\text{оч}}(j)$;
- момент начала обслуживания заявки $t(j)$;
- время $t_{\text{обсл}}(j)$ обслуживания прибором заявки j ;
- момент $t_{\text{коб}}(j)$ окончания обслуживания заявки j и выхода её из СМО.

3) 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				
1				
	$\sum_i R_i(100)$	$\sum_i v_i(100)$	$\sum_i T_i(100)$	$\sum_i \Delta_i(100)$

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	$R_1(100)$	$v_1(100)$	$T_1(100)$	$\Delta_1(100)$
...
	$\sum_i R_i(100)$	$\sum_i v_i(100)$	$\sum_i T_i(100)$	$\sum_i \Delta_i(100)$

где

$R_i(100)$ - число попаданий системы 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 системы в состоянии i на интервале $[0, t_{\text{cob}}(100)]$;

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

4) Для СМО (D|M|1|0) найти:

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}} 100)]$;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}]$;
- Число 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, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}]$;
- Число 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 $[0, t_{\text{cob}}]$;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\bar{z}(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$, 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
$$\bar{t}_{\text{оч}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j);$$
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле
$$\bar{t}_{\text{смo}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_3(j)];$$
- Коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ (отношение времени простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ к $t_{\text{cob}}(100)$).

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.

СМО с отказами - это система массового обслуживания, в которой есть каналы обслуживания, но нет очереди: если заявка приходит, в момент, когда все каналы свободны, то она немедленно обслуживается любым одним каналом, если заявка приходит — когда уже обслуживаются заявки числом меньше, чем число каналов, то она немедленно обслуживается одним из свободных каналов, иначе если заявка приходит — когда заняты

все каналы, то заявка покидает систему (теряется).

Для $(\square/\square/1)$:

— стационарные вероятности состояний:

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

— среднее количество заявок в СМО:

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

— среднего время пребывания в очереди:

$$\overline{t_{\text{оч}}} = \frac{\bar{r}}{\lambda}$$

— среднее время пребывания заявок в СМО:

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

— средняя длина очереди:

$$\bar{r} = \frac{\rho^2}{1 - \rho}$$

Средства языка программирования

В программе использовались следующие функции библиотеки `scipy`:
`scipy.stats.expon.rvs(loc=0, scale= μ , size=100)`, где `size` - размер выборки,
 $\text{scale} = \frac{1}{\lambda} = \mu$

Результаты расчетов

Задание 1

$$V = 14, \Delta T_3 = 0.803, \mu = 1.239$$

Задание 1. Таблица 1:

l	$t_{\text{собр}}(l)$	Type(l)	C(l)	$t_{\text{ост}}(l)$	$t_{\text{ожз}}(l)$	j(l)
1	0.803	1	1	2.14188	0.803	1
2	1.606	2	1	1.33888	0.803	2
3	2.409	2	1	0.53588	0.803	3
4	2.94488	3	0	-1	0.26712	1
5	3.212	1	1	0.30698	0.803	4
6	3.51898	3	0	-1	0.49602	4
7	4.015	1	1	0.16404	0.803	5
8	4.17904	3	0	-1	0.63896	5
9	4.818	1	1	0.18471	0.803	6
10	5.00271	3	0	-1	0.61829	6
11	5.621	1	1	0.67691	0.803	7
12	6.29791	3	0	-1	0.12609	7
13	6.424	1	1	0.7306	0.803	8
14	7.1546	3	0	-1	0.0724	8
15	7.227	1	1	2.69462	0.803	9
16	8.03	2	1	1.89162	0.803	10
17	8.833	2	1	1.08862	0.803	11
18	9.636	2	1	0.28562	0.803	12
19	9.92162	3	0	-1	0.51738	9
20	10.439	1	1	0.85467	0.803	13
21	11.242	2	1	0.05167	0.803	14
22	11.29367	3	0	-1	0.75133	13
23	12.045	1	1	1.11536	0.803	15
24	12.848	2	1	0.31236	0.803	16
25	13.16036	3	0	-1	0.49064	15
26	13.651	1	1	0.85559	0.803	17
27	14.454	2	1	0.05259	0.803	18
28	14.50659	3	0	-1	0.75041	17
29	15.257	1	1	1.11154	0.803	19
30	16.06	2	1	0.30854	0.803	20
31	16.36854	3	0	-1	0.49446	19
32	16.863	1	1	2.62477	0.803	21

33	17.666	2	1	1.82177	0.803	22
34	18.469	2	1	1.01877	0.803	23
35	19.272	2	1	0.21577	0.803	24
36	19.48777	3	0	-1	0.58723	21
37	20.075	1	1	0.0068	0.803	25
38	20.0818	3	0	-1	0.7962	25
39	20.878	1	1	0.09084	0.803	26
40	20.96884	3	0	-1	0.71216	26
41	21.681	1	1	0.28638	0.803	27
42	21.96738	3	0	-1	0.51662	27
43	22.484	1	1	0.86224	0.803	28
44	23.287	2	1	0.05924	0.803	29
45	23.34624	3	0	-1	0.74376	28
46	24.09	1	1	1.33958	0.803	30
47	24.893	2	1	0.53658	0.803	31
48	25.42958	3	0	-1	0.26642	30
49	25.696	1	1	1.66029	0.803	32
50	26.499	2	1	0.85729	0.803	33
51	27.302	2	1	0.05429	0.803	34
52	27.35629	3	0	-1	0.74871	32
53	28.105	1	1	2.69765	0.803	35
54	28.908	2	1	1.89465	0.803	36
55	29.711	2	1	1.09165	0.803	37
56	30.514	2	1	0.28865	0.803	38
57	30.80265	3	0	-1	0.51435	35
58	31.317	1	1	1.03811	0.803	39
59	32.12	2	1	0.23511	0.803	40
60	32.35511	3	0	-1	0.56789	39
61	32.923	1	1	0.83015	0.803	41
62	33.726	2	1	0.02715	0.803	42
63	33.75315	3	0	-1	0.77585	41
64	34.529	1	1	1.02011	0.803	43
65	35.332	2	1	0.21711	0.803	44
66	35.54911	3	0	-1	0.58589	43
67	36.135	1	1	0.50876	0.803	45
68	36.64376	3	0	-1	0.29424	45
69	36.938	1	1	0.31793	0.803	46
70	37.25593	3	0	-1	0.48507	46
71	37.741	1	1	0.46746	0.803	47
72	38.20846	3	0	-1	0.33554	47
73	38.544	1	1	1.05584	0.803	48

74	39.347	2	1	0.25284	0.803	49
75	39.59984	3	0	-1	0.55016	48
76	40.15	1	1	4.1311	0.803	50
77	40.953	2	1	3.3281	0.803	51
78	41.756	2	1	2.5251	0.803	52
79	42.559	2	1	1.7221	0.803	53
80	43.362	2	1	0.9191	0.803	54
81	44.165	2	1	0.1161	0.803	55
82	44.2811	3	0	-1	0.6869	50
83	44.968	1	1	0.91179	0.803	56
84	45.771	2	1	0.10879	0.803	57
85	45.87979	3	0	-1	0.69421	56
86	46.574	1	1	1.26277	0.803	58
87	47.377	2	1	0.45977	0.803	59
88	47.83677	3	0	-1	0.34323	58
89	48.18	1	1	0.15128	0.803	60
90	48.33128	3	0	-1	0.65172	60
91	48.983	1	1	0.02197	0.803	61
92	49.00497	3	0	-1	0.78103	61
93	49.786	1	1	1.30048	0.803	62
94	50.589	2	1	0.49748	0.803	63
95	51.08648	3	0	-1	0.30552	62
96	51.392	1	1	1.88904	0.803	64
97	52.195	2	1	1.08604	0.803	65
98	52.998	2	1	0.28304	0.803	66
99	53.28104	3	0	-1	0.51996	64
100	53.801	1	1	0.02017	0.803	67

Задание 1. Таблица 2:

j	T_3	$t_{обсл}(j)$	$t_{коб}(j)$
1	0.803	2.141881	2.944881
2	1.606	0	1.606
3	2.409	0	2.409
4	3.212	0.306979	3.518979
5	4.015	0.164037	4.179037
6	4.818	0.184714	5.002714
7	5.621	0.676912	6.297912
8	6.424	0.730601	7.154601
9	7.227	2.694618	9.921618
10	8.030	0	8.030
11	8.833	0	8.833
12	9.636	0	9.636

13	10.439	0.854674	11.293674
14	11.242	0	11.242
15	12.045	1.11536	13.16036
16	12.848	0	12.848
17	13.651	0.855588	14.506588
18	14.454	0	14.454
19	15.257	1.111538	16.368538
20	16.060	0	16.060
21	16.863	2.62477	19.48777
22	17.666	0	17.666
23	18.469	0	18.469
24	19.272	0	19.272
25	20.075	0.006799	20.081799
26	20.878	0.090837	20.968837
27	21.681	0.28638	21.96738
28	22.484	0.862235	23.346235
29	23.287	0	23.287
30	24.090	1.339584	25.429584
31	24.893	0	24.893
32	25.696	1.660291	27.356291
33	26.499	0	26.499
34	27.302	0	27.302
35	28.105	2.69765	30.80265
36	28.908	0	28.908
37	29.711	0	29.711
38	30.514	0	30.514
39	31.317	1.038107	32.355107
40	32.12	0	32.12
41	32.923	0.830146	33.753146
42	33.726	0	33.726
43	34.529	1.020107	35.549107
44	35.332	0	35.332
45	36.135	0.508764	36.643764
46	36.938	0.317925	37.255925
47	37.741	0.467461	38.208461
48	38.544	1.055837	39.599837
49	39.347	0	39.347
50	40.15	4.131097	44.281097
51	40.953	0	40.953
52	41.756	0	41.756
53	42.559	0	42.559
54	43.362	0	43.362
55	44.165	0	44.165

56	44.968	0.911793	45.879793
57	45.771	0	45.771
58	46.574	1.26277	47.83677
59	47.377	0	47.377
60	48.18	0.151277	48.331277
61	48.983	0.021966	49.004966
62	49.786	1.300477	51.086477
63	50.589	0	50.589
64	51.392	1.88904	53.28104
65	52.195	0	52.195
66	52.998	0	52.998
67	53.801	0.020166	53.821166

Задание 1. Таблица 3:

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	33	0.33	18.488785	0.343651
1	67	0.67	35.312215	0.656349
	100	1.0	53.80100	1.0

Всего заявок поступило в СМО на интервале: 67

Всего полностью обслуженных заявок: 33

Всего отклонено заявок: 34

Доля отклонённых заявок: 0.492537

Коэффициент простоя прибора: 0.343651

Задание 2

$$V = 14, \Delta T_{06} = 0.796, \lambda = 1.053$$

Задание 2. Таблица 1:

l	$t_{\text{сoб}}(l)$	Type(l)	C(l)	$t_{\text{ост}}(l)$	$t_{\text{ожз}}(l)$	j(l)
1	0.84582	1	1	0.796	0.13971	1
2	0.98554	1	2	0.65629	1.03019	2
3	1.64182	2	1	0.796	0.37391	1
4	2.01573	1	2	0.42209	4.13364	3
5	2.43782	2	1	0.796	3.71154	2
6	3.23382	2	0	0.796	2.91554	3
7	6.14937	1	1	0.796	2.16287	4
8	6.94537	2	0	0.796	1.36687	4
9	8.31224	1	1	0.796	0.82148	5
10	9.10824	2	0	0.796	0.02548	5
11	9.13371	1	1	0.796	0.06371	6
12	9.19743	1	2	0.73229	0.1091	7
13	9.30653	1	3	0.62318	0.56349	8
14	9.87002	1	4	0.05969	0.39241	9
15	9.92971	2	3	0.796	0.33272	6
16	10.26243	1	4	0.46328	0.36972	10
17	10.63215	1	5	0.09356	2.24604	11
18	10.72571	2	4	0.796	2.15248	7
19	11.52171	2	3	0.796	1.35648	8
20	12.31771	2	2	0.796	0.56048	9
21	12.87819	1	3	0.23552	0.10093	12
22	12.97912	1	4	0.13459	1.32192	13
23	13.11371	2	3	0.796	1.18733	10
24	13.90971	2	2	0.796	0.39133	11
25	14.30104	1	3	0.40467	1.66839	14
26	14.70571	2	2	0.796	1.26371	12
27	15.50171	2	1	0.796	0.46771	13
28	15.96943	1	2	0.32829	1.07381	15
29	16.29771	2	1	0.796	0.74552	14
30	17.04323	1	2	0.05048	0.26913	16
31	17.09371	2	1	0.796	0.21864	15
32	17.31236	1	2	0.57736	0.2708	17
33	17.58315	1	3	0.30656	2.00012	18
34	17.88971	2	2	0.796	1.69356	16

35	18.68571	2	1	0.796	0.89756	17
36	19.48171	2	0	0.796	0.10156	18
37	19.58327	1	1	0.796	0.01778	19
38	19.60106	1	2	0.77822	0.08132	20
39	19.68237	1	3	0.6969	0.33547	21
40	20.01784	1	4	0.36143	0.84736	22
41	20.37927	2	3	0.796	0.48593	19
42	20.8652	1	4	0.31007	0.54058	23
43	21.17527	2	3	0.796	0.23051	20
44	21.40578	1	4	0.56549	0.55738	24
45	21.96317	1	5	0.00811	0.10466	25
46	21.97127	2	4	0.796	0.09655	21
47	22.06782	1	5	0.69945	0.27664	26
48	22.34447	1	6	0.4228	0.43533	27
49	22.76727	2	5	0.796	0.01252	22
50	22.77979	1	6	0.78348	0.99954	28
51	23.56327	2	5	0.796	0.21606	23
52	23.77934	1	6	0.57994	0.3165	29
53	24.09584	1	7	0.26344	1.75235	30
54	24.35927	2	6	0.796	1.48891	24
55	25.15527	2	5	0.796	0.69291	25
56	25.84818	1	6	0.10309	1.68122	31
57	25.95127	2	5	0.796	1.57813	26
58	26.74727	2	4	0.796	0.78213	27
59	27.5294	1	5	0.01387	0.15708	32
60	27.54327	2	4	0.796	0.14322	28
61	27.68649	1	5	0.65278	2.0012	33
62	28.33927	2	4	0.796	1.34842	29
63	29.13527	2	3	0.796	0.55242	30
64	29.68769	1	4	0.24358	2.50404	34
65	29.93127	2	3	0.796	2.26046	31
66	30.72727	2	2	0.796	1.46446	32
67	31.52327	2	1	0.796	0.66846	33
68	32.19173	1	2	0.12754	0.2536	35
69	32.31927	2	1	0.796	0.12605	34
70	32.44533	1	2	0.66995	0.55867	36
71	33.004	1	3	0.11128	0.22014	37
72	33.11527	2	2	0.796	0.10887	35
73	33.22414	1	3	0.68713	0.02614	38
74	33.25028	1	4	0.66099	0.27189	39
75	33.52217	1	5	0.3891	0.15366	40

76	33.67583	1	6	0.23544	1.08644	41
77	33.91127	2	5	0.796	0.851	36
78	34.70727	2	4	0.796	0.055	37
79	34.76227	1	5	0.741	0.25324	42
80	35.01551	1	6	0.48776	3.46791	43
81	35.50327	2	5	0.796	2.98015	38
82	36.29927	2	4	0.796	2.18415	39
83	37.09527	2	3	0.796	1.38815	40
84	37.89127	2	2	0.796	0.59215	41
85	38.48342	1	3	0.20385	0.13875	44
86	38.62217	1	4	0.0651	0.26177	45
87	38.68727	2	3	0.796	0.19667	42
88	38.88394	1	4	0.59933	0.20399	46
89	39.08793	1	5	0.39534	0.97845	47
90	39.48327	2	4	0.796	0.5831	43
91	40.06638	1	5	0.2129	0.60537	48
92	40.27927	2	4	0.796	0.39248	44
93	40.67175	1	5	0.40352	1.34793	49
94	41.07527	2	4	0.796	0.94441	45
95	41.87127	2	3	0.796	0.14841	46
96	42.01968	1	4	0.64759	0.40744	50
97	42.42712	1	5	0.24015	0.37526	51
98	42.66727	2	4	0.796	0.13512	47
99	42.80239	1	5	0.66088	0.98875	52
100	43.46327	2	4	0.796	0.32787	48

Задание 2. Таблица 2:

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
1	0.845825	0	0	0.845825	0.796	1.641825
2	0.985536	1	0.656288	1.641825	0.796	2.437825
3	2.015731	1	0.422094	2.437825	0.796	3.233825
4	6.149369	0	0	6.149369	0.796	6.945369
5	8.312238	0	0	8.312238	0.796	9.108238
6	9.133714	0	0	9.133714	0.796	9.929714
7	9.197428	1	0.732286	9.929714	0.796	10.725714
8	9.306532	2	1.419182	10.725714	0.796	11.521714
9	9.870023	3	1.651691	11.521714	0.796	12.317714
10	10.26243	3	2.055284	12.317714	0.796	13.113714
11	10.632152	4	2.481563	13.113714	0.796	13.909714
12	12.878192	2	1.031522	13.909714	0.796	14.705714
13	12.979125	3	1.72659	14.705714	0.796	15.501714
14	14.30104	2	1.200674	15.501714	0.796	16.297714

15	15.969425	1	0.328289	16.297714	0.796	17.093714
16	17.04323	1	0.050484	17.093714	0.796	17.889714
17	17.312357	1	0.577358	17.889714	0.796	18.685714
18	17.583155	2	1.10256	18.685714	0.796	19.481714
19	19.583272	0	0	19.583272	0.796	20.379272
20	19.601056	1	0.778216	20.379272	0.796	21.175272
21	19.682371	2	1.492901	21.175272	0.796	21.971272
22	20.017842	3	1.95343	21.971272	0.796	22.767272
23	20.865201	3	1.902071	22.767272	0.796	23.563272
24	21.405784	3	2.157488	23.563272	0.796	24.359272
25	21.963167	4	2.396106	24.359272	0.796	25.155272
26	22.067824	4	3.087448	25.155272	0.796	25.951272
27	22.344468	5	3.606804	25.951272	0.796	26.747272
28	22.779794	5	3.967478	26.747272	0.796	27.543272
29	23.779335	5	3.763937	27.543272	0.796	28.339272
30	24.095835	6	4.243437	28.339272	0.796	29.135272
31	25.848181	5	3.287091	29.135272	0.796	29.931272
32	27.529403	4	2.401869	29.931272	0.796	30.727272
33	27.686488	4	3.040784	30.727272	0.796	31.523272
34	29.687692	3	1.83558	31.523272	0.796	32.319272
35	32.191727	1	0.127545	32.319272	0.796	33.115272
36	32.445325	1	0.669947	33.115272	0.796	33.911272
37	33.003995	2	0.907277	33.911272	0.796	34.707272
38	33.22414	2	1.483132	34.707272	0.796	35.503272
39	33.25028	3	2.252992	35.503272	0.796	36.299272
40	33.522171	4	2.777101	36.299272	0.796	37.095272
41	33.675829	5	3.419443	37.095272	0.796	37.891272
42	34.76227	4	3.129002	37.891272	0.796	38.687272
43	35.01551	5	3.671762	38.687272	0.796	39.483272
44	38.483422	2	0.999851	39.483272	0.796	40.279272
45	38.622167	3	1.657105	40.279272	0.796	41.075272
46	38.883939	3	2.191333	41.075272	0.796	41.871272
47	39.087931	4	2.783341	41.871272	0.796	42.667272
48	40.066376	4	2.600896	42.667272	0.796	43.463272
49	40.67175	4	-1	-1	-1	-1
50	42.019684	3	-1	-1	-1	-1
51	42.427124	4	-1	-1	-1	-1
52	42.802387	4	-1	-1	-1	-1

Задание 2. Таблица 3:

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	4	0.04	5.255272	0.120913
1	13	0.13	6.005493	0.138174
2	16	0.16	6.645409	0.152897

3	18	0.18	7.117946	0.163769
4	23	0.23	8.356488	0.192266
5	18	0.18	6.67415	0.153558
6	7	0.07	3.145078	0.072362
7	1	0.01	0.263437	0.006061
	100	1.0	43,463273	1.0

Всего заявок поступило в СМО на интервале: 52

Всего полностью обслуженных заявок: 48

Среднее число заявок, находившихся в СМО: 25.62

Среднее время пребывания заявок в очереди: 1.750442

Среднее время пребывания заявок в СМО: 2.546442

Коэффициент простоя прибора: 0.120913

Задание 3

$V = 14$, $\mu = 1.239$, $\lambda = 1.053$

Задание 3. Таблица 1:

l	$t_{\text{собр}}(l)$	Type(l)	C(l)	$t_{\text{ост}}(l)$	$t_{\text{ожз}}(l)$	j(l)
1	0.845825	1	1	2.442066	0.139712	1
2	0.985536	1	2	2.302354	1.030194	2
3	2.015731	1	3	1.272159	4.133638	3
4	3.28789	2	2	2.442066	2.861479	1
5	5.729956	2	1	0.082814	0.419413	2
6	5.812769	2	0	0.272404	0.3366	3
7	6.149369	1	1	0.650561	2.162869	4
8	6.79993	2	0	0.650561	1.512308	4
9	8.312238	1	1	0.121377	0.821477	5
10	8.433615	2	0	0.121377	0.700099	5
11	9.133714	1	1	3.061193	0.063714	6
12	9.197428	1	2	2.997479	0.109104	7
13	9.306532	1	3	2.888375	0.563491	8
14	9.870023	1	4	2.324884	0.392407	9
15	10.26243	1	5	1.932477	0.369722	10
16	10.632152	1	6	1.562755	2.246041	11
17	12.194907	2	5	3.061193	0.683286	6
18	12.878192	1	6	2.377907	0.100932	12
19	12.979125	1	7	2.276975	1.321916	13
20	14.30104	1	8	0.955059	1.668385	14
21	15.256099	2	7	0.473807	0.713326	7
22	15.729906	2	6	3.488891	0.239519	8
23	15.969425	1	7	3.249372	1.073805	15
24	17.04323	1	8	2.175567	0.269126	16
25	17.312357	1	9	1.906441	0.270798	17
26	17.583155	1	10	1.635643	2.000118	18
27	19.218797	2	9	1.004348	0.364475	9
28	19.583272	1	10	0.639873	0.017784	19
29	19.601056	1	11	0.622089	0.081316	20
30	19.682371	1	12	0.540773	0.335471	21
31	20.017842	1	13	0.205303	0.847359	22
32	20.223145	2	12	1.512104	0.642056	10
33	20.865201	1	13	0.870047	0.540583	23
34	21.405784	1	14	0.329464	0.557383	24
35	21.735248	2	13	2.010143	0.227918	11
36	21.963167	1	14	1.782225	0.104657	25
37	22.067824	1	15	1.677568	0.276644	26

38	22.344468	1	16	1.400924	0.435326	27
39	22.779794	1	17	0.965598	0.999542	28
40	23.745391	2	16	1.213168	0.033944	12
41	23.779335	1	17	1.179224	0.3165	29
42	24.095835	1	18	0.862724	1.752346	30
43	24.958559	2	17	0.552008	0.889622	13
44	25.510567	2	16	1.896139	0.337614	14
45	25.848181	1	17	1.558525	1.681221	31
46	27.406706	2	16	0.287864	0.122697	15
47	27.529403	1	17	0.165167	0.157085	32
48	27.686488	1	18	0.008082	2.001204	33
49	27.69457	2	17	2.542143	1.993122	16
50	29.687692	1	18	0.549021	2.504035	34
51	30.236713	2	17	0.849058	1.955014	17
52	31.08577	2	16	0.409437	1.105957	18
53	31.495208	2	15	0.444571	0.696519	19
54	31.939779	2	14	0.551015	0.251948	20
55	32.191727	1	15	0.299066	0.253598	35
56	32.445325	1	16	0.045468	0.55867	36
57	32.490794	2	15	0.763493	0.513202	21
58	33.003995	1	16	0.250291	0.220145	37
59	33.22414	1	17	0.030147	0.02614	38
60	33.25028	1	18	0.004007	0.271892	39
61	33.254286	2	17	1.496201	0.267885	22
62	33.522171	1	18	1.228317	0.153657	40
63	33.675829	1	19	1.074659	1.086441	41
64	34.750488	2	18	1.65799	0.011782	23
65	34.76227	1	19	1.646208	0.25324	42
66	35.01551	1	20	1.392968	3.467911	43
67	36.408478	2	19	0.340064	2.074944	24
68	36.748542	2	18	0.708282	1.73488	25
69	37.456824	2	17	0.390082	1.026598	26
70	37.846906	2	16	2.019507	0.636515	27
71	38.483422	1	17	1.382991	0.138746	44
72	38.622167	1	18	1.244246	0.261772	45
73	38.883939	1	19	0.982474	0.203992	46
74	39.087931	1	20	0.778482	0.978445	47
75	39.866413	2	19	0.608972	0.199963	28
76	40.066376	1	20	0.409009	0.605374	48
77	40.475385	2	19	0.144734	0.196365	29
78	40.620118	2	18	0.065128	0.051631	30
79	40.67175	1	19	0.013496	1.347935	49
80	40.685246	2	18	1.032689	1.334438	31

81	41.717935	2	17	0.163266	0.30175	32
82	41.881201	2	16	1.759345	0.138483	33
83	42.019684	1	17	1.620862	0.40744	50
84	42.427124	1	18	1.213422	0.375263	51
85	42.802387	1	19	0.838158	0.988755	52
86	43.640546	2	18	3.580249	0.150596	34
87	43.791142	1	19	3.429652	1.569127	53
88	45.360269	1	20	1.860526	0.465632	54
89	45.825901	1	21	1.394893	0.276981	55
90	46.102882	1	22	1.117913	1.268253	56
91	47.220794	2	21	1.394044	0.150341	35
92	47.371135	1	22	1.243704	0.535371	57
93	47.906506	1	23	0.708333	1.12202	58
94	48.614839	2	22	0.402452	0.413688	36
95	49.017291	2	21	0.288264	0.011236	37
96	49.028527	1	22	0.277028	0.915328	59
97	49.305554	2	21	1.216023	0.6383	38
98	49.943855	1	22	0.577723	0.584397	60
99	50.521578	2	21	0.091065	0.006674	39
100	50.528252	1	22	0.084391	1.135395	61

Задание 3. Таблица 2:

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
1	0.845825	0	0	0.845825	2.442066	3.28789
2	0.985536	1	4.661606	5.647142	0.082814	5.729956
3	2.015731	2	3.524634	5.540365	0.272404	5.812769
4	6.149369	0	0	6.149369	0.650561	6.79993
5	8.312238	0	0	8.312238	0.121377	8.433615
6	9.133714	0	0	9.133714	3.061193	12.194907
7	9.197428	1	5.584864	14.782292	0.473807	15.256099
8	9.306532	2	2.934484	12.241016	3.488891	15.729906
9	9.870023	3	8.344427	18.21445	1.004348	19.218797
10	10.26243	4	8.448611	18.711041	1.512104	20.223145
11	10.632152	5	9.092954	19.725105	2.010143	21.735248
12	12.878192	5	9.654031	22.532224	1.213168	23.745391
13	12.979125	6	11.427426	24.406551	0.552008	24.958559
14	14.30104	7	9.313389	23.614429	1.896139	25.510567
15	15.969425	6	11.149417	27.118842	0.287864	27.406706
16	17.04323	7	8.109197	25.152427	2.542143	27.69457
17	17.312357	8	12.075298	29.387655	0.849058	30.236713
18	17.583155	9	13.093178	30.676333	0.409437	31.08577
19	19.583272	9	11.467365	31.050637	0.444571	31.495208
20	19.601056	10	11.787709	31.388764	0.551015	31.939779
21	19.682371	11	12.044929	31.727301	0.763493	32.490794

22	20.017842	12	11.740243	31.758085	1.496201	33.254286
23	20.865201	12	12.227297	33.092498	1.65799	34.750488
24	21.405784	13	14.66263	36.068414	0.340064	36.408478
25	21.963167	13	14.077093	36.040259	0.708282	36.748542
26	22.067824	14	14.998918	37.066742	0.390082	37.456824
27	22.344468	15	13.482931	35.827399	2.019507	37.846906
28	22.779794	16	16.477648	39.257441	0.608972	39.866413
29	23.779335	16	16.551315	40.330651	0.144734	40.475385
30	24.095835	17	16.459155	40.554991	0.065128	40.620118
31	25.848181	16	13.804376	39.652557	1.032689	40.685246
32	27.529403	16	14.025265	41.554668	0.163266	41.717935
33	27.686488	17	12.435368	40.121856	1.759345	41.881201
34	29.687692	17	10.372606	40.060297	3.580249	43.640546
35	32.191727	14	13.635023	45.82675	1.394044	47.220794
36	32.445325	15	15.767061	48.212387	0.402452	48.614839
37	33.003995	15	15.725032	48.729027	0.288264	49.017291
38	33.22414	16	14.865391	48.089531	1.216023	49.305554
39	33.25028	17	17.180233	50.430512	0.091065	50.521578
40	33.522171	17	16.732458	50.254629	0.358014	50.612643
41	33.675829	18	-1	-1	-1	-1
42	34.76227	18	-1	-1	-1	-1
43	35.01551	19	-1	-1	-1	-1
44	38.483422	16	-1	-1	-1	-1
45	38.622167	17	-1	-1	-1	-1
46	38.883939	18	-1	-1	-1	-1
47	39.087931	19	-1	-1	-1	-1
48	40.066376	19	-1	-1	-1	-1
49	40.67175	18	-1	-1	-1	-1
50	42.019684	16	-1	-1	-1	-1
51	42.427124	17	-1	-1	-1	-1
52	42.802387	18	-1	-1	-1	-1
53	43.791142	18	-1	-1	-1	-1
54	45.360269	19	-1	-1	-1	-1
55	45.825901	20	-1	-1	-1	-1
56	46.102882	21	-1	-1	-1	-1
57	47.371135	21	-1	-1	-1	-1
58	47.906506	22	-1	-1	-1	-1
59	49.028527	21	-1	-1	-1	-1
60	49.943855	21	-1	-1	-1	-1
61	50.528252	21	-1	-1	-1	-1

Задание 3. Таблица 3:

Состояние	$R_i(100)$	$\nu_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	3	0.03	3.394831	0.067187

1	5	0.05	1.058178	0.020942
2	3	0.03	3.581364	0.070878
3	2	0.02	1.83565	0.036329
4	1	0.01	0.392407	0.007766
5	2	0.02	1.053007	0.02084
6	3	0.03	1.903206	0.037666
7	3	0.03	2.869528	0.056791
8	2	0.02	1.224185	0.024228
9	2	0.02	0.635273	0.012573
10	2	0.02	1.653426	0.032723
11	1	0.01	0.081316	0.001609
12	2	0.02	0.977527	0.019346
13	3	0.03	0.973804	0.019272
14	3	0.03	0.68607	0.013578
15	4	0.04	1.488015	0.029449
16	9	0.09	2.37963	0.047095
17	13	0.13	7.785454	0.154081
18	12	0.12	4.169506	0.082518
19	9	0.09	4.637433	0.091779
20	4	0.04	3.046091	0.060285
21	5	0.05	1.083532	0.021444
22	6	0.06	2.910487	0.057601
23	1	0.01	0.708333	0.014019
	100	1	50,528253	0.999999

Всего заявок поступило в СМО на интервале: 61

Всего полностью обслуженных заявок: 39

Среднее число заявок, находившихся в СМО: 29.04

Среднее время пребывания заявок в очереди: 10.543618

Среднее время пребывания заявок в СМО: 11.620207

Коэффициент простоя прибора: 0.067187

Список литературы

1. Системы массового обслуживания [Электронный ресурс]: методические указания / А.А. Лобузов. - М.: РТУ МИРЭА, 2023. - 45 с.
2. Аникина О., Гущина О. М. Табличное моделирование динамики работы одноканальной системы массового обслуживания с ограниченной очередью. – Тольятти, 2017. – 12 с.
3. Плескунов М. А. Теория массового обслуживания. – Екатеринбург: Уральский энергетический институт, 2022. – 268 с.

Приложение

Код выполнения работы:

```
import scipy.stats as sps
import numpy as np

V = 14
precision = 6
N = 100

import numpy as np
import csv

def export_table(arr, file_name):
    precision = 6

    # Write data to a CSV file
    with open(f"{file_name}.csv", mode='w', newline='') as file:
        writer = csv.writer(file)
        for row in arr:
            row = [f"{np.round(x, precision - 1)}" for x in row]
            writer.writerow(row)

# (D|M|1|0)
def task1():
    t_demand = 0.803
    mu = 1.239
    l = 1 #processing demand number
    l_j = 1 #current demand number
    l_print = 1 #printable num of demand
    k = 1 #quantity of served demands
    t_event = t_demand #moment of event execution
    current_event_stat = 1 #demand is recieved and is serving
    current_sys_stat = 1 #system is serving a demand
    np.random.seed(12345)
    times_of_services = sps.expon.rvs(scale=1/mu, size=N)
    remaining_serv_time = times_of_services[0]
    remaining_demand_time = t_demand
    table_1 = []
    table_2 = [None] * N
    table_2[0] = [1, 1 * t_demand, times_of_services[k - 1], 1 * t_demand +
times_of_services[k - 1]]
    table_3 = [[0,0,0,0], [0,0,0,0]]
    for j in range(2, 102):
        table_1.append([j-1, t_event, current_event_stat, current_sys_stat,
remaining_serv_time, remaining_demand_time, l_print])
        #time to serve less then time to recieve new demand
        if remaining_serv_time < remaining_demand_time and current_sys_stat ==
1:
            t_event = t_event + remaining_serv_time
            current_event_stat = 3
            current_sys_stat = 0
            remaining_demand_time -= remaining_serv_time
            remaining_serv_time = -1
            l_print = 1

        elif remaining_serv_time > remaining_demand_time:
            t_event += remaining_demand_time
```

```

        current_event_stat = 2
        current_sys_stat = 1
        remaining_serv_time -= remaining_demand_time
        remaining_demand_time = t_demand
        l_j += 1
        l_print = l_j
        table_2[l_j - 1] = [l_j, t_event, 0, t_event]

    elif current_sys_stat == 0:
        if j != 101:
            table_3[0][2] += remaining_demand_time #time of waiting
            t_event += remaining_demand_time
            current_event_stat = 1
            current_sys_stat = 1
            remaining_serv_time = times_of_services[k]
            remaining_demand_time = t_demand
            l_j += 1
            l = l_j
            l_print = l_j
            k += 1
            table_2[l - 1] = [l, l * t_demand, times_of_services[k - 1], l *
t_demand + times_of_services[k - 1]]

    max_demand_num = 0
    for arr in table_1:
        if arr[-1] > max_demand_num:
            max_demand_num = arr[-1]
    table_2 = table_2[0:max_demand_num]

    for row in table_1:
        table_3[row[3]][0] += 1
    table_3[0][2] += t_demand
    table_3[1][2] = table_1[-1][1] - table_3[0][2]

    table_3[0][1] = table_3[0][0]/(table_3[0][0] + table_3[1][0])
    table_3[1][1] = table_3[1][0]/(table_3[0][0] + table_3[1][0])

    table_3[0][3] = table_3[0][2]/(table_3[0][2] + table_3[1][2])
    table_3[1][3] = table_3[1][2]/(table_3[0][2] + table_3[1][2])

    return table_1, table_2, table_3

t1, t2, t3 = task1()
t3[0].insert(0, 0)
t3[1].insert(0, 1)
export_table(t1, "table_11")
export_table(t2, "table_12")
export_table(t3, "table_13")

print(f"Count of demands: {len(t2)}")
print(f"Count of unserved demands: {sum(map((lambda x: x[2] == 0), t2))}")

#(M|D|1|0)
def task2():
    t_serv = 0.796
    e_lambda = 1.053
    l = 1 #processing demand number
    l_j = 1 #current demand number
    l_print = 1 #printable num of demand

```

```

k = 1 #quantity of served demands
current_event_stat = 1 #demand is recieved and is serving
current_sys_stat = 1 #system is serving a demand
np.random.seed(2048)
times_of_demandes = sps.expon.rvs(scale=1/e_lambda, size=N)
remaining_demand_time = times_of_demandes[1]
remaining_serv_time = t_serv
t_event = times_of_demandes[0] #moment of event execution
q = [1]
task = 1
table_1 = []
table_2 = [None] * N
table_2[0] = [1, t_event, 0, 0, t_event, t_serv, t_event + t_serv]
table_3 = [[0, 0, 0, 0]]
table_3[0][2] = t_event

for j in range(2, 102):
    table_1.append([j-1, t_event, current_event_stat, current_sys_stat,
remaining_serv_time, remaining_demand_time, l_print])
    #time to serve less then time to recieve new demand
    if remaining_serv_time <= remaining_demand_time and current_sys_stat >
0:
        if j != 101:
            if len(table_3) < current_sys_stat+1:
                table_3.append([0, 0, 0, 0])
            table_3[current_sys_stat][2] += remaining_serv_time

            t_event = t_event + remaining_serv_time
            current_event_stat = 2
            current_sys_stat -= 1
            task = q.pop(0)
            l_print = task
            t_serv_start = t_event - t_serv
            table_2[task - 1][3] = t_serv_start - table_2[task - 1][1]
            if table_2[task - 1][3] < 0.000001:
                table_2[task - 1][3] = 0
            table_2[task - 1][4] = t_serv_start
            table_2[task - 1][5] = t_serv
            table_2[task - 1][6] = t_event

            remaining_demand_time -= remaining_serv_time
            remaining_serv_time = t_serv

        elif remaining_serv_time > remaining_demand_time or current_sys_stat
== 0:
            if j != 101:
                if len(table_3) < current_sys_stat+1:
                    table_3.append([0, 0, 0, 0])
                table_3[current_sys_stat][2] += remaining_demand_time

            t_event += remaining_demand_time
            current_event_stat = 1
            if current_sys_stat == 0:
                remaining_serv_time = t_serv
            else:
                remaining_serv_time -= remaining_demand_time
            current_sys_stat += 1
            l_j += 1
            remaining_demand_time = times_of_demandes[l_j]
            q.append(l_j)
            l_print = l_j

```

```

        table_2[l_j - 1] = [l_j, t_event, current_sys_stat-1, -1, -1, -1, -
1]

max_demand_num = 0
for arr in table_1:
    if arr[-1] > max_demand_num:
        max_demand_num = arr[-1]
table_2 = table_2[0:max_demand_num]

#if table_1[-1][2] == 1
for row in table_1:
    table_3[row[3]][0] += 1

for i in range(len(table_3)):
    table_3[i][1] = table_3[i][0] / 100
    table_3[i][3] = table_3[i][2] / table_1[-1][1]

    return table_1, table_2, table_3
t1, t2, t3 = task2()

for i in range(len(t3)):
    t3[i].insert(0, i)
export_table(t1, "table_21")
export_table(t2, "table_22")
export_table(t3, "table_23")

#(M|M|1|0)
def task3():
    mu = 1.239
    e_lambda = 1.053
    l = 1 # processing demand number
    l_j = 1 # current demand number
    l_print = 1 # printable num of demand
    current_event_stat = 1 # demand is recieved and is serving
    current_sys_stat = 1 # system is serving a demand
    np.random.seed(2048)
    times_of_demandes = sps.expon.rvs(scale=1/e_lambda, size=N)
    remaining_demand_time = times_of_demandes[1]
    times_of_services = sps.expon.rvs(scale=1/mu, size=N)
    remaining_serv_time = times_of_services[0]
    t_event = times_of_demandes[0] # moment of event execution
    q = [1]
    task = 1
    table_1 = []
    table_2 = [None] * N
    table_2[0] = [1, t_event, 0, -1, -1, -1, -1]
    table_3 = [[0, 0, 0, 0]]
    table_3[0][2] = t_event

    for j in range(2, 102):
        table_1.append([j-1, t_event, current_event_stat, current_sys_stat,
remaining_serv_time, remaining_demand_time, l_print])
        #time to serve less then time to recieve new demand
        if remaining_serv_time <= remaining_demand_time and current_sys_stat >
0:
            if len(table_3) < current_sys_stat+1:
                table_3.append([0, 0, 0, 0])

```

```

        if j != 101:
            table_3[current_sys_stat][2] += remaining_serv_time

            t_event = t_event + remaining_serv_time
            current_event_stat = 2
            current_sys_stat -= 1
            task = q.pop(0)
            t_serv = times_of_services[task-1]
            l_print = task
            t_serv_start = t_event - t_serv
            table_2[task - 1][3] = t_serv_start - table_2[task - 1][1]
            if table_2[task - 1][3] < 0.000001:
                table_2[task - 1][3] = 0
            table_2[task - 1][4] = t_serv_start
            table_2[task - 1][5] = t_serv
            table_2[task - 1][6] = t_event

            remaining_demand_time -= remaining_serv_time
            remaining_serv_time = t_serv

    elif remaining_serv_time > remaining_demand_time or current_sys_stat
== 0:
        if len(table_3) < current_sys_stat+1:
            table_3.append([0, 0, 0, 0])
        if j != 101:
            table_3[current_sys_stat][2] += remaining_demand_time

            t_event += remaining_demand_time
            current_event_stat = 1
            l_j += 1
            if current_sys_stat == 0:
                remaining_serv_time = times_of_services[l_j-1]
            else:
                remaining_serv_time -= remaining_demand_time
            current_sys_stat += 1
            remaining_demand_time = times_of_demandes[l_j]
            q.append(l_j)
            l_print = l_j
            table_2[l_j - 1] = [l_j, t_event, current_sys_stat-1, -1, -1, -1, -
1]

max_demand_num = 0
for arr in table_1:
    if arr[-1] > max_demand_num:
        max_demand_num = arr[-1]
table_2 = table_2[0:max_demand_num]

for row in table_1:
    table_3[row[3]][0] += 1

for i in range(len(table_3)):
    table_3[i][1] = table_3[i][0] / 100
    table_3[i][3] = table_3[i][2] / table_1[-1][1]

    return table_1, table_2, table_3

t1, t2, t3 = task3()

for i in range(len(t3)):

```



```

        t3[i].insert(0, i)
    export_table(t1, "table_31")
    export_table(t2, "table_32")
    export_table(t3, "table_33")

    print(f"Count of demands: {len(t2)}")
    print(f"Count of unserved demands: {sum(map((lambda x: x[3] != -1),
    t2))}")
    z = 0
    for i in range(100):
        cnt = 0
        t_event = t1[i][1]
        for j in range(len(t2)):
            if t2[j][1] >= t_event:
                cnt += 1
        z += cnt
    print(f"z(100): {z/100}")

    sum_q_time = 0
    sum_stay_time = 0
    for i in range(serv_demands):
        sum_q_time += t2[i][3]
        sum_stay_time += (t2[i][6] - t2[i][1])
    q_time_mean = sum_q_time / serv_demands
    print(f"q_time_mean: {round(q_time_mean, 5)}")
    stay_time_mean = sum_stay_time / serv_demands
    print(f"stay_time_mean: {round(stay_time_mean, 5)}")
    print(f"downtime: {round(t3[0][3] / t1[-1][1], 5)}")

    print(f"Count of demands: {len(t2)}")
    serv_demands = sum(map((lambda x: x[2] == 2), t1))
    print(f"Count of served demands: {serv_demands}")
    z = 0
    for i in range(100):
        cnt = 0
        t_event = t1[i][1]
        for j in range(len(t2)):
            if t2[j][1] >= t_event:
                cnt += 1
        z += cnt
    print(f"z(100): {z/100}")

    sum_q_time = 0
    sum_stay_time = 0
    for i in range(serv_demands):
        sum_q_time += t2[i][3]
        sum_stay_time += (t2[i][6] - t2[i][1])
    q_time_mean = sum_q_time / serv_demands
    print(f"q_time_mean: {round(q_time_mean, 5)}")
    stay_time_mean = sum_stay_time / serv_demands
    print(f"stay_time_mean: {round(stay_time_mean, 5)}")
    print(f"downtime: {round(t3[0][3] / t1[-1][1], 5)}")

```