



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

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА  
КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

**Типовой расчет**  
по дисциплине **«Системы массового обслуживания»**

**ВАРИАНТ 90**

Выполнил:  
Студент 4-го курса  
Демченко Г. Д.

Группа: КМБО-04-21

МОСКВА 2024

## Оглавление

Задание.....	3
Краткие теоретические сведения.....	16
Результаты расчетов.....	21
Задача 1.....	21
Задание 2.....	28
Задание 3.....	35
Задача 4.....	41
Задание 5.....	47
Список литературы.....	54
Приложение.....	55

## Задание

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени  $t$  характеризуется числом заявок, находящихся в СМО. События в развитии СМО связаны либо с поступлением в неё новых заявок, либо с окончанием обслуживания прибором заявки. В момент поступления в СМО очередной заявки определяется в соответствии с условием Задания время, через которое в СМО поступит следующая заявка. Время обслуживания прибором заявки определяется в соответствии с условием Задания в момент поступления заявки в прибор.

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

#### Дано:

- время между приходом заявок  $\Delta T_z$  (заданная постоянная величина);
- параметр  $\mu_1$  показательного распределения времени обслуживания заявки прибором.

#### СМО имеется 2 состояния:

- 0 – в системе нет заявок (прибор свободен),
- 1 – в системе одна заявка (прибор занят).

#### События могут быть трех 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 – завершение обслуживания заявки прибором (при этом СМО переходит из состояния 1 в состояние 0);
- 3 – появление в СМО новой заявки, которая получает отказ в обслуживании (прибор занят, при этом СМО остается в состоянии 1).

Предполагается, что в начальный момент времени  $t=0$  в СМО нет заявок, т.е. состояние системы 0, и через заданное время  $\Delta T_z$  в СМО поступает первая заявка (произойдет событие с номером 1). Момент наступления первого события (типа 1) равен  $t_{\text{cob}}(1)=\Delta T_z$ . 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  $\mu_1$ .

#### Требуется:

Провести моделирование первых 100 событий в развитии СМО и составить следующие таблицы.

Таблица 1.1 с данными о событиях:

- номер события  $l$  ;
- момент наступления события  $t_{\text{cob}}(l)$ ;
- тип события  $Type(l)$ ;

- состояние СМО  $C(l)$  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  $l$  .

Таблица 1.2 с данными о 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  $t_{коб}(j)=t_3(j)$ .

Таблица 1.3 с данными о состояниях следующего 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 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  $[0, t_{cob}(100)]$  ;

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

А также найти:

- число заявок  $J(100)$  , поступивших в СМО на интервале  $[0, t_{cob}(100)]$  ;
- число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$  ;
- число  $JL(100)$  отклоненных заявок на интервале  $[0, t_{cob}(100)]$  ;
- общее время занятости прибора на интервале  $[0, t_{cob}(100)]$  ;
- общее время простоя прибора на интервале  $[0, t_{cob}(100)]$ .

**Задание 2.** Одноканальная СМО с бесконечной очередью ( $M \vee D \vee 1$ )

**Дано:**

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

СМО имеется бесконечное число состояний: 0 – в системе нет заявок, 1 – в системе одна заявка, и т.д. События могут быть двух типов:

- 1 – появление в СМО новой заявки;
- 2 – завершение обслуживания заявки прибором.

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  $\lambda$ .

### **Требуется:**

Провести моделирование первых 100 событий в развитии СМО и составить следующие таблицы.

Таблица 2.1 с данными о событиях:

- номер события  $l$ ;
- момент наступления события  $t_{cob}(l)$ ;
- тип события  $Type(l)$ ;
- состояние СМО  $C(l)$  после события  $l$ ;
- оставшееся время  $t_{ocm}(l)$  обслуживания прибором заявки после события  $l$  (если после события прибор свободен, то  $t_{ocm}(l)=-1$ );
- время ожидания  $t_{ожз}(l)$ , через которое после события  $l$  в СМО появится новая заявка;
- номер заявки  $j(l)$ , участвующей в событии  $l$ .

Таблица 2.2 с данными о 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 временно заносится:

$$t_{оч}(j)=t_{ноб}(j)=t_{обсл}(j)=t_{коб}(j)=-1.$$

Настоящие значения заносятся позже по мере их определения.

Таблица 2.3 с данными о состояниях следующего 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  $[0, t_{cob}(100)]$ ;

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

А также найти:

- число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ ;
- число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ ;

- среднее число заявок, находившихся в СМО, на интервале  $[0, t_{\text{cob}}(100)]$ , которое находится по формуле  $z(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$ , 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  $t_{\text{оч}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j)$ ;
- среднее время пребывания заявок в СМО на интервале  $[0, t_{\text{cob}}(100)]$ , которое находится по формуле  $t_{\text{СМО}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_z(j)]$ ;
- общее время простоя прибора на интервале  $[0, t_{\text{cob}}(100)]$ .

**Задание 3.** Одноканальная СМО с бесконечной очередью ( $M \vee M \vee 1$ )  
**Дано:**

- среднее число заявок, поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром  $\lambda$ );
- параметр  $\mu_1$  показательного распределения времени обслуживания заявки прибором.

СМО имеет бесконечное число состояний: 0 – в системе нет заявок, 1 – в системе одна заявка, и т.д. События могут быть двух типов:

- 1 – появление в СМО новой заявки;
- 2 – завершение 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  $t_z(1)$  в соответствии с показательным законом распределения с параметром  $\lambda$ , а в момент поступления каждой заявки на обслуживание в прибор определяется время её обслуживания  $t_{\text{обсл}}(1)$  в соответствии с показательным законом распределения с параметром  $\mu_1$ .

**Требуется:**

Провести моделирование первых 100 событий в развитии СМО и составить следующие таблицы.

Таблица 3.1 с данными о событиях:

- номер события  $l$ ;
- 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  $l$  в СМО появится новая заявка;
- номер заявки  $j(l)$ , участвующей в событии  $l$ .

Таблица 3.2 с данными о заявках:

- номер заявки  $j$ ;
- момент  $t_z(j)$  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  $t_{\text{ноб}}(j)$ ;
- 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 занят, и она становится в очередь, то в таблицу временно заносится:

$$t_{\text{оч}}(j) = t_{\text{ноб}}(j) = t_{\text{обсл}}(j) = t_{\text{коб}}(j) = -1.$$

Настоящие значения заносятся позже по мере их определения.

Таблица 3.3 с данными о состояниях следующего 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  $[0, t_{\text{cob}}(100)]$ ;

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

А также 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  $z(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$ , 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  $t_{\text{оч}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j)$ ;

- среднее время пребывания заявок в СМО на интервале  $[0, t_{\text{cob}}(100)]$ , которое

находится по формуле  $t_{\text{СМО}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_z(j)]$ ;

- общее время простоя прибора на интервале  $[0, t_{\text{cob}}(100)]$ .

**Задание 4.** Система массового обслуживания ( $M \vee M \vee n$ )

**Дано:**

- число приборов  $n$  ;

- среднее число заявок  $\lambda$ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром  $\lambda$ );
- параметр  $\mu_2$  показательного распределения времени обслуживания заявки каждым прибором.

Все приборы пронумерованы. СМО имеет бесконечное число состояний: 0 – в системе нет заявок, 1 – в системе одна заявка, и т.д. События могут быть двух типов: 1 – появление в системе новой заявки, 2 – завершение обслуживания заявки прибором (при этом данный прибор освобождается, и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в этот прибор). Если при появлении в системе новой заявки есть свободные приборы, то она сразу же принимается на обслуживание свободным прибором с наименьшим номером, в противном случае заявка становится в очередь.

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  $\lambda$ , 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  $\mu_2$ .

### **Требуется:**

Провести моделирование первых 100 событий в развитии СМО и составить следующие таблицы.

Таблица 4.1 с данными о событиях:

- номер события  $l$ ;
- момент наступления события  $t_{cob}(l)$ ;
- тип события  $Type(l)$ ;
- состояние СМО  $C(l)$  после события  $l$ ;
- минимальное оставшееся время  $t_{осмин}(l)$  обслуживания прибором заявки после события  $l$  (если после события прибор свободен, то  $t_{осмин}(l)=-1$ );

- время ожидания  $t_{ожз}(l)$ , через которое после события  $l$  в СМО появится новая заявка;
- номер заявки  $j(l)$ , участвующей в событии  $l$ .
- номер прибора  $k(l)$ , участвующем в событии  $l$  (если заявка встала в очередь, то  $k(l)=-1$ )

Таблица 4.2 с данными о 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 временно заносится:

$$t_{оч}(j)=t_{ноб}(j)=t_{обсл}(j)=t_{коб}(j)=-1.$$

Настоящие значения заносятся позже по мере их определения.

Таблица 4.3 с данными о приборах следующего вида:

$k$	$N(k)$	$t_{зан}(k)$	$t_{np}(k)$	$\Delta_{np}(k)$
$l$	$N(l)$	$t_{зан}(l)$	$t_{np}(l)$	$\Delta_{np}(l)$
...	...	...	...	...
$n$	$N(n)$	$t_{зан}(n)$	$t_{np}(n)$	$\Delta_{np}(n)$
	$\sum_{k=0}^n N(k)$	$\frac{1}{n} \sum_{k=0}^n t_{зан}(k)$	$\frac{1}{n} \sum_{k=0}^n t_{np}(k)$	$\frac{1}{n} \sum_{k=0}^n \Delta_{np}(k)$

где

$k$  -номер прибора;

$N(k)$  -общее число заявок, поступивших на обслуживание в прибор  $k$  на

интервале  $[0, t_{\text{cob}}(100)]$ ;

$t_{\text{зан}}(k)$  - общее время занятости прибора  $k$  на интервале  $[0, t_{\text{cob}}(100)]$ ;

$t_{\text{np}}(k)$  - общее время простоя прибора  $k$  на интервале  $[0, t_{\text{cob}}(100)]$ ;

$\Delta_{\text{np}}(k) = \frac{t_{\text{np}}(k)}{t_{\text{cob}}(100)}$  - коэффициент простоя прибора  $k$  на интервале  $[0, t_{\text{cob}}(100)]$ .

А также 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  $\bar{z}(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$ , 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  $\bar{t}_{\text{СМО}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_3(j)]$ ;

#### Задание 5. Система массового обслуживания ( $M \vee M \vee n \vee m$ )

**Дано:**

- число приборов  $n$ ;
- число мест в очереди  $m$ ;
- среднее число заявок  $\lambda$ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром  $\lambda$ );
- параметр  $\mu_2$  показательного распределения времени обслуживания заявки каждым прибором.

Все приборы пронумерованы. СМО имеет конечное число состояний:

0 – в системе нет заявок, 1 – в системе одна заявка, ...,  $(n+m)$  – в системе  $(n+m)$  заявок (все приборы заняты и в очереди нет свободных мест). События могут быть трех типов:

1 – появление в системе новой заявки;

2 – завершение обслуживания заявки прибором (при этом данный прибор освобождается, и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в этот прибор);

3 – появление в СМО новой заявки, которая получает отказ в обслуживании (все приборы заняты и в очереди нет свободных мест, при этом СМО остается в состоянии  $(n+m)$ ).

Если при появлении в системе новой заявки есть свободные приборы, то она сразу же принимается на обслуживание свободным прибором с наименьшим номером, в противном случае заявка становится в очередь (если там есть свободные места).

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  $\lambda$ , 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  $\mu_2$ .

### **Требуется:**

Провести моделирование первых 100 событий в развитии СМО и составить следующие таблицы.

Таблица 5.1 с данными о событиях:

- номер события  $l$ ;
- момент наступления события  $t_{соб}(l)$ ;
- тип события  $Type(l)$ ;
- состояние СМО  $C(l)$  после события  $l$ ;
- минимальное оставшееся время  $t_{осмин}(l)$  обслуживания прибором заявки после события  $l$  (если после события прибор свободен, то  $t_{осмин}(l)=-1$ );
- время ожидания  $t_{ожсз}(l)$ , через которое после события  $l$  в СМО появится новая заявка;

- номер заявки  $j(l)$ , участвующей в событии  $l$ .
- номер прибора  $k(l)$ , участвующем в событии  $l$  (если заявка встала в очередь, то  $k(l)=-1$ ).

Таблица 5.2 с данными о 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  $q(j)=0$ , если заявка получила отказ в обслуживании, то  $q(j)=-1$ );
- время пребывания заявки в очереди  $t_{oc}(j)$  (если заявка получила отказ в обслуживании, то  $t_{oc}(j)=0$ );
- момент начала обслуживания заявки  $t_{ноб}(j)$  (если заявка получила отказ в обслуживании, то  $t_{ноб}(j)=-1$ );
- время обслуживания заявки  $t_{обсл}(j)$  (если заявка получила отказ в обслуживании, то  $t_{обсл}(j)=0$ );
- момент  $t_{коб}(j)$  окончания обслуживания заявки  $j$  и выхода её из СМО (если заявка получила отказ в обслуживании, то  $t_{коб}(j)=t_3(j)$ );
- номер прибора  $k(j)$ , который обслуживал заявку  $j$  (если заявка получила отказ в обслуживании, то  $k(j)=-2$ ).

Если в момент появления заявки  $j$  в СМО прибор занят, и она становится в очередь, то в таблицу временно заносится:

$$t_{oc}(j)=t_{ноб}(j)=t_{обсл}(j)=t_{коб}(j)=-1.$$

Настоящие значения заносятся позже по мере их определения.

Таблица 5.3 с данными о приборах следующего вида:

$k$	$N(k)$	$t_{зан}(k)$	$t_{np}(k)$	$\Delta_{np}(k)$
$l$	$N(l)$	$t_{зан}(l)$	$t_{np}(l)$	$\Delta_{np}(l)$
...	...	...	...	...

$n$	$N(n)$	$t_{зан}(n)$	$t_{np}(n)$	$\Delta_{np}(n)$
	$\sum_{k=0}^n N(k)$	$\frac{1}{n} \sum_{k=0}^n t_{зан}(k)$	$\frac{1}{n} \sum_{k=0}^n t_{np}(k)$	$\frac{1}{n} \sum_{k=0}^n \Delta_{np}(k)$

где

$k$  -номер прибора;

$N(k)$  -общее число заявок, поступивших на обслуживание в прибор  $k$  на интервале  $[0, t_{cob}(100)]$ ;

$t_{зан}(k)$  -общее время занятости прибора  $k$  на интервале  $[0, t_{cob}(100)]$ ;

$t_{np}(k)$  -общее время простоя прибора  $k$  на интервале  $[0, t_{cob}(100)]$ ;

$\Delta_{np}(k) = \frac{t_{np}(k)}{t_{cob}(100)}$  - коэффициент простоя прибора  $k$  на интервале  $[0, t_{cob}(100)]$ .

А также найти:

- число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ ;
- число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ ;
- среднее число заявок, находившихся в СМО, на интервале  $[0, t_{cob}(100)]$ ,

которое находится по формуле  $z(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$ , где  $z(l)$  – число заявок в СМО после события  $l$ ;

- среднее время пребывания заявок в очереди на интервале  $[0, t_{cob}(100)]$ ,

которое находится по формуле  $t_{оч}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{оч}(j)$ ;

- среднее время пребывания заявок в СМО на интервале  $[0, t_{cob}(100)]$ , которое

находится по формуле  $t_{СМО}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{коб}(j) - t_3(j)]$ ;

**Вывод результатов проводить с округлением до 0,00001.**

## Краткие теоретические сведения

**Система массового обслуживания (СМО)** — математическая модель систем, предназначенных для обслуживания заявок (требований, запросов, клиентов, заказчиков и т.д.) поступающих в неё, как правило, в случайные моменты времени.

**Одноканальная СМО с очередью** — это система массового обслуживания, где предусмотрены места в очереди. Если заявка поступает, когда канал занят, она не получает отказа, а становится в очередь и дожидается освобождения канала, который сможет её обслужить. На вход одноканальной СМО с  $m$ -очередью поступает поток заявок с интенсивностью  $\lambda$ . Интенсивность потока обслуживания канала равна  $\mu$ . Если заявка приходит, когда канал свободен, она принимается на обслуживание, и канал занимается ею. После завершения обслуживания канал освобождается. Если заявка приходит, когда канал занят, она встает в очередь и ждет обслуживания. Дисциплина очереди стандартная: кто пришел первым, тот обслуживается. Максимальная длина очереди —  $m$ . Если заявка приходит, когда мест в очереди  $m$  заявок нет, она получает отказ и исключается из обслуживания.

**Многоканальные системы массового обслуживания** — это системы, в которых несколько обслуживающих каналов (серверов) одновременно предоставляют услуги поступающим требованиям или клиентам. Такие системы широко применяются для моделирования процессов обслуживания в телекоммуникациях, транспорте, сфере услуг, производстве и других областях.

### 1) Системы с неограниченной очередью

Характеристики:

- Неограниченная очередь: Число мест в очереди не ограничено. Любое число клиентов может ожидать обслуживания.
- Обслуживание: Если все каналы заняты, поступившие клиенты становятся в очередь и ждут освобождения любого из каналов.
- Отсутствие потерь: Поскольку очередь неограничена, ни один клиент не теряется из системы из-за невозможности занять место в очереди.

Модель:

- Обозначение:  $M/M/c$  (по классификации Кендалла).
  - $M$  (Markovian) — поступление требований происходит по простейшему (пуассоновскому) потоку.
  - $M$  — время обслуживания имеет экспоненциальное распределение.



-  $c$  — число обслуживающих каналов.

## 2) Системы с ограниченной очередью

Характеристики:

- Ограниченная очередь: Существует фиксированная максимальная длина очереди ( $K$ ). Если очередь заполнена, новые клиенты не могут попасть в систему.
- Обслуживание: Клиенты, прибывающие в систему, когда очередь полна, получают отказ и покидают систему.
- Наличие потерь: Из-за ограниченной вместимости часть клиентов теряется, что может быть критично для некоторых систем.

Модель:

- Обозначение:  $M/M/c/K$ .
  - $K$  — максимальное число клиентов в системе (включая обслуживаемых и ожидающих).

### **СМО M|M|1**

Для СМО M|M|1 в стационарном режиме характеристики определяются следующим формулами:

#### **Стационарные вероятности состояний:**

$$P_n = (1 - \rho) \rho^n, \quad n \geq 0,$$

где коэффициент загрузки системы  $\rho = \frac{\lambda}{\mu}$ ,  $\lambda$  – интенсивность входящего потока заявок,  $\mu$  – интенсивность обслуживания.

#### **Средняя длина очереди:**

Среднее число заявок в очереди (не считая обслуживаемую) определяется как:

$$L_q = \frac{\rho^2}{1 - \rho}$$

#### **Среднее время пребывания заявок в очереди:**

Среднее время ожидания заявки в очереди вычисляется по формуле:

$$W_q = \frac{L_q}{\lambda} = \frac{\rho}{\mu(1 - \rho)}$$

#### **Среднее время пребывания заявок в системе:**

Среднее полное время, которое заявка проводит в системе (очередь плюс обслуживание), составляет:

$$W = W_q + \frac{1}{\mu} = \frac{1}{\mu - \lambda} = \frac{1}{\mu(1 - \rho)}$$

## СМО M|M|n

### Стационарные вероятности состояний:

Вероятность того, что в системе находится k заявок, вычисляется по формуле:

Для  $0 \leq k < n$ :

$$P_k = \frac{a^k}{k!} P_0,$$

Для  $k \geq n$ :

$$P_k = \frac{a^k}{n! n^{k-n}} P_0,$$

где:

$$a = \frac{\lambda}{\mu} - \text{параметр нагрузки,}$$

$\lambda$  – интенсивность входящего потока заявок,

$\mu$  – интенсивность обслуживания одной заявки

Вероятность отсутствия заявок в системе  $P_0$  определяется из условия нормировки:

$$P_0 = \left[ \sum_{k=0}^{n-1} \frac{a^k}{k!} + \frac{a^n}{n!} \frac{1}{1-\rho} \right]^{-1}$$

где  $\rho = \frac{a}{n} = \frac{\lambda}{n\mu}$  – коэффициент загрузки системы, при условии  $\rho < 1$

### Среднее число занятых приборов:

$$L_s = \frac{\lambda}{\mu} = a.$$

### Средняя длина очереди:

$$L_q = \frac{P_{ож} \rho}{1-\rho}$$

где вероятность ожидания в очереди  $P_{ож}$  определяется по формуле Эрланга:

$$P_{ож} = \frac{\frac{a^n}{n!} \frac{n}{n-a}}{\sum_{k=0}^{n-1} \frac{a^k}{k!} + \frac{a^n}{n!} \frac{n}{n-a}}$$

### Среднее время пребывания заявок в очереди:

$$W_q = \frac{L_q}{\lambda} = \frac{P_{ож}}{\mu(1-\rho)}$$

### Среднее время пребывания заявок в системе:

$$W = W_q + \frac{1}{\mu} = \frac{1}{\mu} + \frac{P_{ож}}{\mu(1-\rho)}$$

## СМО M|M|n|m

### Стационарные вероятности состояний:

Вероятность того, что в системе находится  $k$  заявок, определяется по формуле:

Для  $0 \leq k \leq n$ :

$$P_k = \frac{a^k}{k!} P_0$$

Для  $n \leq k \leq m$ :

$$P_k = \frac{a^k}{n! n^{k-n}} P_0$$

Вероятность отсутствия заявок в системе  $P_0$  вычисляется как:

$$P_0 = \left[ \sum_{k=0}^{n-1} \frac{a^k}{k!} + \frac{a^n}{n!} \sum_{k=n}^m \left( \frac{a}{n} \right)^{k-n} \right]^{-1}$$

### Среднее число занятых приборов:

$$L_s = \sum_{k=1}^n k P_k + n \sum_{k=n+1}^m P_k$$

### Средняя длина очереди:

$$L_q = \sum_{k=n+1}^m (k-n) P_k$$

### Среднее время пребывания заявок в очереди:

$$W_q = \frac{L_q}{\lambda_{\text{эфф}}},$$

где эффективная интенсивность входящего потока:

$$\lambda_{\text{эфф}} = \lambda (1 - P_{\text{отк}}),$$

а вероятность отказа в обслуживании  $P_{\text{отк}} = P_m$ .

### Среднее время пребывания заявок в системе:

$$W = W_q + \frac{1}{\mu}$$

### Вероятность отказа в обслуживании:

$$P_{\text{отк}} = P_m = \frac{a^m}{n! n^{m-n}} P_0$$

## Результаты расчетов

### Задача 1

СМО (D|M|1|0):

$V=90$ ,  $\Delta T_3=0.724$ ,  $\mu_1=1.412$

Таблица 1.1 Данные о событиях

$l$	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$T_{\text{ocm}}(l)$	$t_{\text{ожсз}}(l)$	$j(l)$
1	0.72400	1	1	0.24244	0.72400	1
2	0.96644	2	0	-1.00000	0.48156	1
3	1.44800	1	1	0.14912	0.72400	2
4	1.59712	2	0	-1.00000	0.57488	2
5	2.17200	1	1	0.30387	0.72400	3
6	2.47587	2	0	-1.00000	0.42013	3
7	2.89600	1	1	0.44746	0.72400	4
8	3.34346	2	0	-1.00000	0.27654	4
9	3.62000	1	1	0.09675	0.72400	5
10	3.71675	2	0	-1.00000	0.62725	5
11	4.34400	1	1	2.53211	0.72400	6
12	5.06800	3	1	1.80811	0.72400	7
13	5.79200	3	1	1.08411	0.72400	8
14	6.51600	3	1	0.36011	0.72400	9
15	6.87611	2	0	-1.00000	0.36389	6
16	7.24000	1	1	0.16018	0.72400	10
17	7.40018	2	0	-1.00000	0.56382	10
18	7.96400	1	1	0.39085	0.72400	11
19	8.35485	2	0	-1.00000	0.33315	11
20	8.68800	1	1	0.89801	0.72400	12
21	9.41200	3	1	0.17401	0.72400	13
22	9.58601	2	0	-1.00000	0.54999	12
23	10.13600	1	1	0.01978	0.72400	14

24	10.15578	2	0	-1.00000	0.70422	14
25	10.86000	1	1	0.42636	0.72400	15
26	11.28636	2	0	-1.00000	0.29764	15
27	11.58400	1	1	1.24423	0.72400	16
28	12.30800	3	1	0.52023	0.72400	17
29	12.82823	2	0	-1.00000	0.20377	16
30	13.03200	1	1	0.57054	0.72400	18
31	13.60254	2	0	-1.00000	0.15346	18
32	13.75600	1	1	0.18458	0.72400	19
33	13.94058	2	0	-1.00000	0.53942	19
34	14.48000	1	1	0.03873	0.72400	20
35	14.51873	2	0	-1.00000	0.68527	20
36	15.20400	1	1	0.00432	0.72400	21
37	15.20832	2	0	-1.00000	0.71968	21
38	15.92800	1	1	1.47945	0.72400	22
39	16.65200	3	1	0.75545	0.72400	23
40	17.37600	3	1	0.03145	0.72400	24
41	17.40745	2	0	-1.00000	0.69255	22
42	18.10000	1	1	2.14678	0.72400	25
43	18.82400	3	1	1.42278	0.72400	26
44	19.54800	3	1	0.69878	0.72400	27
45	20.24678	2	0	-1.00000	0.02522	25
46	20.27200	1	1	1.18790	0.72400	28
47	20.99600	3	1	0.46390	0.72400	29
48	21.45990	2	0	-1.00000	0.26010	28
49	21.72000	1	1	1.69887	0.72400	30
50	22.44400	3	1	0.97487	0.72400	31
51	23.16800	3	1	0.25087	0.72400	32
52	23.41887	2	0	-1.00000	0.47313	30
53	23.89200	1	1	0.40025	0.72400	33

54	24.29225	2	0	-1.00000	0.32375	33
55	24.61600	1	1	0.57984	0.72400	34
56	25.19584	2	0	-1.00000	0.14416	34
57	25.34000	1	1	0.06323	0.72400	35
58	25.40323	2	0	-1.00000	0.66077	35
59	26.06400	1	1	0.22913	0.72400	36
60	26.29313	2	0	-1.00000	0.49487	36
61	26.78800	1	1	1.45236	0.72400	37
62	27.51200	3	1	0.72836	0.72400	38
63	28.23600	3	1	0.00436	0.72400	39
64	28.24036	2	0	-1.00000	0.71964	37
65	28.96000	1	1	1.94188	0.72400	40
66	29.68400	3	1	1.21788	0.72400	41
67	30.40800	3	1	0.49388	0.72400	42
68	30.90188	2	0	-1.00000	0.23012	40
69	31.13200	1	1	0.61165	0.72400	43
70	31.74365	2	0	-1.00000	0.11235	43
71	31.85600	1	1	0.52520	0.72400	44
72	32.38120	2	0	-1.00000	0.19880	44
73	32.58000	1	1	0.47589	0.72400	45
74	33.05589	2	0	-1.00000	0.24811	45
75	33.30400	1	1	0.18465	0.72400	46
76	33.48865	2	0	-1.00000	0.53935	46
77	34.02800	1	1	0.39718	0.72400	47
78	34.42518	2	0	-1.00000	0.32682	47
79	34.75200	1	1	0.15758	0.72400	48
80	34.90958	2	0	-1.00000	0.56642	48
81	35.47600	1	1	0.56934	0.72400	49
82	36.04534	2	0	-1.00000	0.15466	49
83	36.20000	1	1	0.02581	0.72400	50

84	36.22581	2	0	-1.00000	0.69819	50
85	36.92400	1	1	0.38499	0.72400	51
86	37.30899	2	0	-1.00000	0.33901	51
87	37.64800	1	1	0.88963	0.72400	52
88	38.37200	3	1	0.16563	0.72400	53
89	38.53763	2	0	-1.00000	0.55837	52
90	39.09600	1	1	0.13813	0.72400	54
91	39.23413	2	0	-1.00000	0.58587	54
92	39.82000	1	1	0.33918	0.72400	55
93	40.15918	2	0	-1.00000	0.38482	55
94	40.54400	1	1	0.02664	0.72400	56
95	40.57064	2	0	-1.00000	0.69736	56
96	41.26800	1	1	0.10601	0.72400	57
97	41.37401	2	0	-1.00000	0.61799	57
98	41.99200	1	1	0.82256	0.72400	58
99	42.71600	3	1	0.09856	0.72400	59
100	42.81456	2	0	-1.00000	0.62544	58

Таблица 1.2 Данные о заявках

$j$	$t_z(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
0	0.72400	0.24244	0.96644
1	1.44800	0.14912	1.59712
2	2.17200	0.30387	2.47587
3	2.89600	0.44746	3.34346
4	3.62000	0.09675	3.71675
5	4.34400	2.53211	6.87611
6	5.06800	0.00000	5.06800
7	5.79200	0.00000	5.79200
8	6.51600	0.00000	6.51600
9	7.24000	0.16018	7.40018



10	7.96400	0.39085	8.35485
11	8.68800	0.89801	9.58601
12	9.41200	0.00000	9.41200
13	10.13600	0.01978	10.15578
14	10.86000	0.42636	11.28636
15	11.58400	1.24423	12.82823
16	12.30800	0.00000	12.30800
17	13.03200	0.57054	13.60254
18	13.75600	0.18458	13.94058
19	14.48000	0.03873	14.51873
20	15.20400	0.00432	15.20832
21	15.92800	1.47945	17.40745
22	16.65200	0.00000	16.65200
23	17.37600	0.00000	17.37600
24	18.10000	2.14678	20.24678
25	18.82400	0.00000	18.82400
26	19.54800	0.00000	19.54800
27	20.27200	1.18790	21.45990
28	20.99600	0.00000	20.99600
29	21.72000	1.69887	23.41887
30	22.44400	0.00000	22.44400
31	23.16800	0.00000	23.16800
32	23.89200	0.40025	24.29225
33	24.61600	0.57984	25.19584
34	25.34000	0.06323	25.40323
35	26.06400	0.22913	26.29313
36	26.78800	1.45236	28.24036
37	27.51200	0.00000	27.51200
38	28.23600	0.00000	28.23600
39	28.96000	1.94188	30.90188

40	29.68400	0.00000	29.68400
41	30.40800	0.00000	30.40800
42	31.13200	0.61165	31.74365
43	31.85600	0.52520	32.38120
44	32.58000	0.47589	33.05589
45	33.30400	0.18465	33.48865
46	34.02800	0.39718	34.42518
47	34.75200	0.15758	34.90958
48	35.47600	0.56934	36.04534
49	36.20000	0.02581	36.22581
50	36.92400	0.38499	37.30899
51	37.64800	0.88963	38.53763
52	38.37200	0.00000	38.37200
53	39.09600	0.13813	39.23413
54	39.82000	0.33918	40.15918
55	40.54400	0.02664	40.57064
56	41.26800	0.10601	41.37401
57	41.99200	0.82256	42.81456
58	42.71600	0.00000	42.71600

Таблица 1.3 Данные о состояниях

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	41	0.41	7.01776	0.16391
1	59	0.59	35.7968	0.83609
	100	1.0	42.81456	1.0

Число заявок  $J(100)$  , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ : **59**

Число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ : **41**

Число  $JL(100)$  отклоненных заявок на интервале  $[0, t_{cob}(100)]$ : **18**

Общее время занятости прибора на интервале  $[0, t_{cob}(100)]$ : **7.01776**

Общее время простоя прибора на интервале  $[0, t_{cob}(100)]$ : **35.7968**

## Задание 2

СМО (M|D|1):

$V=90$ ,  $T_{об}=0.738$ ,  $\lambda=1.142$

Таблица 2.1 Данные о событиях

$l$	$t_{cob}(l)$	$Type(l)$	$C(l)$	$T_{ocm}(l)$	$t_{ожсз}(l)$	$j(l)$
1	0.98812	1	1	0.73800	1.35642	1
2	1.72612	2	0	-1.00000	0.61842	1
3	2.34454	1	1	0.73800	0.69839	2
4	3.04293	1	2	0.03961	0.64898	3
5	3.78093	2	1	0.73800	0.60937	2
6	4.39030	1	2	0.12863	1.67548	4
7	5.12830	2	1	0.73800	1.54685	3
8	5.86630	2	0	-1.00000	0.80885	4
9	6.67515	1	1	0.73800	1.20516	5
10	7.41315	2	0	-1.00000	0.46716	5
11	7.88031	1	1	0.73800	0.06309	6
12	7.94340	1	2	0.67491	1.62189	7
13	8.68140	2	1	0.73800	0.94698	6
14	9.41940	2	0	-1.00000	0.20898	7
15	9.62838	1	1	0.73800	1.93122	8
16	10.36638	2	0	-1.00000	1.19322	8
17	11.55960	1	1	0.73800	0.73103	9
18	12.29063	1	2	0.00697	1.39276	10
19	13.02863	2	1	0.73800	1.38579	9
20	13.76663	2	0	-1.00000	0.64779	10
21	14.41442	1	1	0.73800	0.84801	11
22	15.15242	2	0	-1.00000	0.11001	11
23	15.26243	1	1	0.73800	1.95930	12
24	16.00043	2	0	-1.00000	1.22130	12

25	17.22173	1	1	0.73800	2.16750	13
26	17.95973	2	0	-1.00000	1.42950	13
27	19.38923	1	1	0.73800	2.60724	14
28	20.12723	2	0	-1.00000	1.86924	14
29	21.99647	1	1	0.73800	2.37029	15
30	22.73447	2	0	-1.00000	1.63229	15
31	24.36676	1	1	0.73800	1.24283	16
32	25.10476	2	0	-1.00000	0.50483	16
33	25.60959	1	1	0.73800	2.26732	17
34	26.34759	2	0	-1.00000	1.52932	17
35	27.87691	1	1	0.73800	0.91420	18
36	28.61491	2	0	-1.00000	0.17620	18
37	28.79111	1	1	0.73800	2.03444	19
38	29.52911	2	0	-1.00000	1.29644	19
39	30.82555	1	1	0.73800	0.58773	20
40	31.41328	1	2	0.15027	2.62461	21
41	32.15128	2	1	0.73800	2.47434	20
42	32.88928	2	0	-1.00000	1.73634	21
43	34.62562	1	1	0.73800	0.15546	22
44	34.78108	1	2	0.58254	1.59009	23
45	35.51908	2	1	0.73800	1.00755	22
46	36.25708	2	0	-1.00000	0.26955	23
47	36.52663	1	1	0.73800	0.57898	24
48	37.10561	1	2	0.15902	0.75644	25
49	37.84361	2	1	0.73800	0.59742	24
50	38.44103	1	2	0.14058	1.55426	26
51	39.17903	2	1	0.73800	1.41368	25
52	39.91703	2	0	-1.00000	0.67568	26
53	40.59271	1	1	0.73800	0.45565	27
54	41.04836	1	2	0.28235	0.48863	28

55	41.78636	2	1	0.73800	0.20628	27
56	41.99264	1	2	0.53172	3.12288	29
57	42.73064	2	1	0.73800	2.59116	28
58	43.46864	2	0	-1.00000	1.85316	29
59	45.32180	1	1	0.73800	2.33927	30
60	46.05980	2	0	-1.00000	1.60127	30
61	47.66107	1	1	0.73800	1.04696	31
62	48.39907	2	0	-1.00000	0.30896	31
63	48.70803	1	1	0.73800	0.38242	32
64	49.09045	1	2	0.35558	1.93885	33
65	49.82845	2	1	0.73800	1.58327	32
66	50.56645	2	0	-1.00000	0.84527	33
67	51.41172	1	1	0.73800	0.39511	34
68	51.80683	1	2	0.34289	0.19327	35
69	52.00010	1	3	0.14962	0.29450	36
70	52.73810	2	2	0.73800	0.14488	34
71	52.88298	1	3	0.59312	2.14262	37
72	53.62098	2	2	0.73800	1.54950	35
73	54.35898	2	1	0.73800	0.81150	36
74	55.09698	2	0	-1.00000	0.07350	37
75	55.17048	1	1	0.73800	1.30269	38
76	55.90848	2	0	-1.00000	0.56469	38
77	56.47317	1	1	0.73800	0.23414	39
78	56.70731	1	2	0.50386	0.20229	40
79	56.90960	1	3	0.30157	1.21278	41
80	57.64760	2	2	0.73800	0.91121	39
81	58.38560	2	1	0.73800	0.17321	40
82	58.55881	1	2	0.56479	1.51845	42
83	59.29681	2	1	0.73800	0.95366	41
84	60.03481	2	0	-1.00000	0.21566	42

85	60.25047	1	1	0.73800	0.85458	43
86	60.98847	2	0	-1.00000	0.11658	43
87	61.10505	1	1	0.73800	0.20507	44
88	61.31012	1	2	0.53293	3.81658	45
89	62.04812	2	1	0.73800	3.28365	44
90	62.78612	2	0	-1.00000	2.54565	45
91	65.33177	1	1	0.73800	2.64157	46
92	66.06977	2	0	-1.00000	1.90357	46
93	67.97334	1	1	0.73800	1.06586	47
94	68.71134	2	0	-1.00000	0.32786	47
95	69.03920	1	1	0.73800	0.53883	48
96	69.57803	1	2	0.19917	7.49662	49
97	70.31603	2	1	0.73800	7.29745	48
98	71.05403	2	0	-1.00000	6.55945	49
99	77.61348	1	1	0.73800	0.30444	50
100	77.91792	1	2	0.43356	0.62026	51

Таблица 2.2 Данные о заявках

$j$	$t_z(j)$	$q(j)$	$t_{oc}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$T_{коб}(j)$
1	0.98812	0	0.00000	0.98812	0.73800	1.72612
2	2.34454	0	0.00000	2.34454	0.73800	3.08254
3	3.04293	1	0.77761	3.82054	0.73800	4.55854
4	4.39030	1	0.86663	5.25693	0.73800	5.99493
5	6.67515	0	0.00000	6.67515	0.73800	7.41315
6	7.88031	0	0.00000	7.88031	0.73800	8.61831
7	7.94340	1	1.41291	9.35631	0.73800	10.09431
8	9.62838	0	0.00000	9.62838	0.73800	10.36638
9	11.55960	0	0.00000	11.55960	0.73800	12.29760
10	12.29063	1	0.74497	13.03560	0.73800	13.77360
11	14.41442	0	0.00000	14.41442	0.73800	15.15242

12	15.26243	0	0.00000	15.26243	0.73800	16.00043
13	17.22173	0	0.00000	17.22173	0.73800	17.95973
14	19.38923	0	0.00000	19.38923	0.73800	20.12723
15	21.99647	0	0.00000	21.99647	0.73800	22.73447
16	24.36676	0	0.00000	24.36676	0.73800	25.10476
17	25.60959	0	0.00000	25.60959	0.73800	26.34759
18	27.87691	0	0.00000	27.87691	0.73800	28.61491
19	28.79111	0	0.00000	28.79111	0.73800	29.52911
20	30.82555	0	0.00000	30.82555	0.73800	31.56355
21	31.41328	1	0.88827	32.30155	0.73800	33.03955
22	34.62562	0	0.00000	34.62562	0.73800	35.36362
23	34.78108	1	1.32054	36.10162	0.73800	36.83962
24	36.52663	0	0.00000	36.52663	0.73800	37.26463
25	37.10561	1	0.89702	38.00263	0.73800	38.74063
26	38.44103	1	0.87858	39.31961	0.73800	40.05761
27	40.59271	0	0.00000	40.59271	0.73800	41.33071
28	41.04836	1	1.02035	42.06871	0.73800	42.80671
29	41.99264	1	1.26972	43.26236	0.73800	44.00036
30	45.32180	0	0.00000	45.32180	0.73800	46.05980
31	47.66107	0	0.00000	47.66107	0.73800	48.39907
32	48.70803	0	0.00000	48.70803	0.73800	49.44603
33	49.09045	1	1.09358	50.18403	0.73800	50.92203
34	51.41172	0	0.00000	51.41172	0.73800	52.14972
35	51.80683	1	1.08089	52.88772	0.73800	53.62572
36	52.00010	2	1.62562	53.62572	0.73800	54.36372
37	52.88298	2	2.06912	54.95210	0.73800	55.69010
38	55.17048	0	0.00000	55.17048	0.73800	55.90848
39	56.47317	0	0.00000	56.47317	0.73800	57.21117
40	56.70731	1	1.24186	57.94917	0.73800	58.68717
41	56.90960	2	1.77757	58.68717	0.73800	59.42517



42	58.55881	1	1.30279	59.86160	0.73800	60.59960
43	60.25047	0	0.00000	60.25047	0.73800	60.98847
44	61.10505	0	0.00000	61.10505	0.73800	61.84305
45	61.31012	1	1.27093	62.58105	0.73800	63.31905
46	65.33177	0	0.00000	65.33177	0.73800	66.06977
47	67.97334	0	0.00000	67.97334	0.73800	68.71134
48	69.03920	0	0.00000	69.03920	0.73800	69.77720
49	69.57803	1	0.93717	70.51520	0.73800	71.25320
50	77.61348	0	0.00000	77.61348	0.73800	78.35148
51	77.91792	1	1.17156	79.08948	0.73800	79.82748

Таблица 2.3 Данные о состояниях

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	30	0.30000	33.31074	0.42751
1	47	0.47000	29.05662	0.37291
2	20	0.20000	12.34844	0.15848
3	3	0.03000	2.21400	0.02841
	100	1	76.93000	1.00000

Число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ :

**51**

Число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ :

**49**

Среднее число заявок, находившихся в СМО, на интервале  $[0, t_{cob}(100)]$ :

**0.96**

Среднее время пребывания заявок в очереди на интервале  $[0, t_{cob}(100)]$ :

**0.48261**

Среднее время пребывания заявок в СМО на интервале  $[0, t_{cob}(100)]$ :

**1.25073**

Общее время простоя прибора на интервале  $[0, t_{cob}(100)]$ :

**33.31074**

### Задание 3

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

$V=90$ ,  $\lambda=1.142$ ,  $\mu_1=1.412$

Таблица 3.1 Данные о событиях

$l$	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ocm}}(l)$	$t_{\text{ожсз}}(l)$	$j(l)$
1	0.98892	1	1	0.50663	0.01569	1
2	1.00461	1	2	0.49094	1.16557	2
3	1.49555	2	1	0.67739	0.67463	1
4	2.17018	1	2	0.00276	0.26296	3
5	2.17294	2	1	0.77358	0.26020	2
6	2.43314	1	2	0.51338	0.63180	4
7	2.94652	2	1	0.46601	0.11842	3
8	3.06494	1	2	0.34759	1.03051	5
9	3.41253	2	1	1.94090	0.68292	4
10	4.09545	1	2	1.25798	1.17574	6
11	5.27119	1	3	0.08224	0.17989	7
12	5.35343	2	2	0.74123	0.09765	5
13	5.45108	1	3	0.64358	0.70165	8
14	6.09466	2	2	0.49853	0.05807	6
15	6.15273	1	3	0.44046	0.80816	9
16	6.59319	2	2	0.87854	0.36770	7
17	6.96089	1	3	0.51084	0.04795	10
18	7.00884	1	4	0.46289	0.81526	11
19	7.47173	2	3	1.05670	0.35237	8
20	7.82410	1	4	0.70433	1.11830	12
21	8.52843	2	3	0.24827	0.41397	9
22	8.77670	2	2	0.27116	0.16570	10
23	8.94240	1	3	0.10546	0.28168	13
24	9.04786	2	2	0.28483	0.17622	11
25	9.22408	1	3	0.10861	0.48241	14

26	9.33269	2	2	0.15419	0.37380	12
27	9.48688	2	1	1.68067	0.21961	13
28	9.70649	1	2	1.46106	0.46648	15
29	10.17297	1	3	0.99458	0.79703	16
30	10.97000	1	4	0.19755	1.00718	17
31	11.16755	2	3	0.60048	0.80963	14
32	11.76803	2	2	0.89969	0.20915	15
33	11.97718	1	3	0.69054	0.27667	18
34	12.25385	1	4	0.41387	1.27190	19
35	12.66772	2	3	0.20413	0.85803	16
36	12.87185	2	2	0.20122	0.65390	17
37	13.07307	2	1	0.12671	0.45268	18
38	13.19978	2	0	-1.00000	0.32597	19
39	13.52575	1	1	0.28636	0.03106	20
40	13.55681	1	2	0.25530	0.00867	21
41	13.56548	1	3	0.24663	1.38255	22
42	13.81211	2	2	0.40855	1.13592	20
43	14.22066	2	1	0.09008	0.72737	21
44	14.31074	2	0	-1.00000	0.63729	22
45	14.94803	1	1	2.05208	0.61011	23
46	15.55814	1	2	1.44197	3.12413	24
47	17.00011	2	1	0.96388	1.68216	23
48	17.96399	2	0	-1.00000	0.71828	24
49	18.68227	1	1	0.67121	0.90229	25
50	19.35348	2	0	-1.00000	0.23108	25
51	19.58456	1	1	0.19714	0.06465	26
52	19.64921	1	2	0.13249	2.15174	27
53	19.78170	2	1	0.65568	2.01925	26
54	20.43738	2	0	-1.00000	1.36357	27
55	21.80095	1	1	0.46856	0.38087	28

56	22.18182	1	2	0.08769	0.85835	29
57	22.26951	2	1	0.84719	0.77066	28
58	23.04017	1	2	0.07653	0.42123	30
59	23.11670	2	1	1.74950	0.34470	29
60	23.46140	1	2	1.40480	1.60496	31
61	24.86620	2	1	1.20238	0.20016	30
62	25.06636	1	2	1.00222	0.94814	32
63	26.01450	1	3	0.05408	1.12176	33
64	26.06858	2	2	1.33319	1.06768	31
65	27.13626	1	3	0.26551	0.50465	34
66	27.40177	2	2	0.99909	0.23914	32
67	27.64091	1	3	0.75995	0.25616	35
68	27.89707	1	4	0.50379	2.03485	36
69	28.40086	2	3	0.81128	1.53106	33
70	29.21214	2	2	0.85710	0.71978	34
71	29.93192	1	3	0.13732	0.35873	37
72	30.06924	2	2	0.73880	0.22141	35
73	30.29065	1	3	0.51739	0.35376	38
74	30.64441	1	4	0.16363	0.26821	39
75	30.80804	2	3	0.08321	0.10458	36
76	30.89125	2	2	0.65976	0.02137	37
77	30.91262	1	3	0.63839	0.33662	40
78	31.24924	1	4	0.30177	0.88684	41
79	31.55101	2	3	0.76458	0.58507	38
80	32.13608	1	4	0.17951	0.77610	42
81	32.31559	2	3	0.16414	0.59659	39
82	32.47973	2	2	0.19930	0.43245	40
83	32.67903	2	1	0.58985	0.23315	41
84	32.91218	1	2	0.35670	0.21572	43
85	33.12790	1	3	0.14098	0.21553	44

86	33.26888	2	2	0.33612	0.07455	42
87	33.34343	1	3	0.26157	0.51394	45
88	33.60500	2	2	0.55410	0.25237	43
89	33.85737	1	3	0.30173	0.12546	46
90	33.98283	1	4	0.17627	0.59544	47
91	34.15910	2	3	0.33862	0.41917	44
92	34.49772	2	2	0.20772	0.08055	45
93	34.57827	1	3	0.12717	3.47171	48
94	34.70544	2	2	1.13019	3.34454	46
95	35.83563	2	1	0.17822	2.21435	47
96	36.01385	2	0	-1.00000	2.03613	48
97	38.04998	1	1	0.94753	0.91315	49
98	38.96313	1	2	0.03438	0.49096	50
99	38.99751	2	1	2.52519	0.45658	49
100	39.45409	1	2	2.06861	0.18523	51

Таблица 3.2 Данные о заявках

$j$	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$
1	0.98892	0	0.00000	0.98892	0.50663	1.49555
2	1.00461	1	0.49094	1.49555	0.67739	2.17294
3	2.17018	1	0.00276	2.17294	0.77358	2.94652
4	2.43314	1	0.51338	2.94652	0.46601	3.41253
5	3.06494	1	0.34759	3.41253	1.94090	5.35343
6	4.09545	1	1.25798	5.35343	0.74123	6.09466
7	5.27119	2	0.82347	6.09466	0.49853	6.59319
8	5.45108	2	1.14211	6.59319	0.87854	7.47173
9	6.15273	2	1.31900	7.47173	1.05670	8.52843
10	6.96089	2	1.56754	8.52843	0.24827	8.77670
11	7.00884	3	1.76786	8.77670	0.27116	9.04786
12	7.82410	3	1.22376	9.04786	0.28483	9.33269
13	8.94240	2	0.39029	9.33269	0.15419	9.48688

14	9.22408	2	0.26280	9.48688	1.68067	11.16755
15	9.70649	1	1.46106	11.16755	0.60048	11.76803
16	10.17297	2	1.59506	11.76803	0.89969	12.66772
17	10.97000	3	1.69772	12.66772	0.20413	12.87185
18	11.97718	2	0.89467	12.87185	0.20122	13.07307
19	12.25385	3	0.81922	13.07307	0.12671	13.19978
20	13.52575	0	0.00000	13.52575	0.28636	13.81211
21	13.55681	1	0.25530	13.81211	0.40855	14.22066
22	13.56548	2	0.65518	14.22066	0.09008	14.31074
23	14.94803	0	0.00000	14.94803	2.05208	17.00011
24	15.55814	1	1.44197	17.00011	0.96388	17.96399
25	18.68227	0	0.00000	18.68227	0.67121	19.35348
26	19.58456	0	0.00000	19.58456	0.19714	19.78170
27	19.64921	1	0.13249	19.78170	0.65568	20.43738
28	21.80095	0	0.00000	21.80095	0.46856	22.26951
29	22.18182	1	0.08769	22.26951	0.84719	23.11670
30	23.04017	1	0.07653	23.11670	1.74950	24.86620
31	23.46140	1	1.40480	24.86620	1.20238	26.06858
32	25.06636	1	1.00222	26.06858	1.33319	27.40177
33	26.01450	2	1.38727	27.40177	0.99909	28.40086
34	27.13626	2	1.26460	28.40086	0.81128	29.21214
35	27.64091	2	1.57123	29.21214	0.85710	30.06924
36	27.89707	3	2.17217	30.06924	0.73880	30.80804
37	29.93192	2	0.87612	30.80804	0.08321	30.89125
38	30.29065	2	0.60060	30.89125	0.65976	31.55101
39	30.64441	3	0.90660	31.55101	0.76458	32.31559
40	30.91262	2	1.40297	32.31559	0.16414	32.47973
41	31.24924	3	1.23049	32.47973	0.19930	32.67903
42	32.13608	3	0.54295	32.67903	0.58985	33.26888
43	32.91218	1	0.35670	33.26888	0.33612	33.60500

44	33.12790	2	0.47710	33.60500	0.55410	34.15910
45	33.34343	2	0.81567	34.15910	0.33862	34.49772
46	33.85737	2	0.64035	34.49772	0.20772	34.70544
47	33.98283	3	0.72261	34.70544	1.13019	35.83563
48	34.57827	2	1.25736	35.83563	0.17822	36.01385
49	38.04998	0	0.00000	38.04998	0.94753	38.99751
50	38.96313	1	0.03438	38.99751	-0.03438	38.96313
51	39.45409	1	-1.00000	-1.00000	-1.00000	-1.00000

Таблица 3.3 Данные о состояниях

Состояние	$R_i(100)$	$v_i(100)$	$T_i(100)$	$\Delta_i(100)$
0	6	0.06000	5.31232	0.13465
1	22	0.22000	8.66234	0.21955
2	35	0.35000	13.19207	0.33437
3	28	0.28000	8.19483	0.20771
4	9	0.09000	3.10361	0.07866
	100	1	38.46517	1.00000

число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ :

**51**

число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ :

**49**

среднее число заявок, находившихся в СМО, на интервале  $[0, t_{cob}(100)]$ :

**2.12**

среднее время пребывания заявок в очереди на интервале  $[0, t_{cob}(100)]$ :

**0.77332**

среднее время пребывания заявок в СМО на интервале  $[0, t_{cob}(100)]$ :

**0.71555**

общее время простоя прибора на интервале  $[0, t_{cob}(100)]$ :

**5.31232**



### Задача 4

Система массового обслуживания (M|M|n)

$V=90$ ,  $\lambda=1.142$ ,  $\mu_2=0.235$ ,  $n=6$

Таблица 4.1 Данные о событиях

$l$	$t_{\text{соб}}(l)$	$Type(l)$	$C(l)$	$t_{\text{осмин}}(l)$	$t_{\text{ож}}(j)$	$j(l)$	$k(l)$
1	0.51505	1	1	2.16997	2.01969	1	1
2	2.53474	1	2	0.15028	0.18327	2	2
3	2.68502	2	1	6.31112	0.03299	1	1
4	2.71801	1	2	0.58301	0.36869	3	1
5	3.08670	1	3	0.21432	0.67065	4	3
6	3.30102	2	2	2.63973	0.45633	3	1
7	3.75735	1	3	2.18340	0.86732	5	1
8	4.62467	1	4	1.31608	0.89269	6	4
9	5.51736	1	5	0.42339	0.88410	7	5
10	5.94075	2	4	2.84209	0.46071	4	3
11	6.40146	1	5	1.09370	0.90844	8	3
12	7.30990	1	6	0.03471	0.13201	9	6
13	7.34461	2	5	0.15055	0.09730	9	6
14	7.44191	1	6	0.05325	0.36103	10	6
15	7.49516	2	5	1.23211	0.30778	8	3
16	7.80294	1	6	0.92433	0.71601	11	3
17	8.51895	1	7	0.20832	1.14060	12	-1
18	8.72727	2	6	0.05557	0.93228	10	6
19	8.78284	2	5	0.21330	0.87671	6	4
20	8.99614	2	4	0.09459	0.66341	2	2
21	9.09073	2	3	3.49981	0.56882	5	1
22	9.65955	1	4	0.33144	1.08360	13	1
23	9.99099	2	3	2.59955	0.75216	13	1
24	10.74315	1	4	1.84739	1.19365	14	1
25	11.93680	1	5	0.65374	0.18697	15	2

26	12.12377	1	6	0.46677	0.27409	16	4
27	12.39786	1	7	0.19268	0.10592	17	6
28	12.50378	1	8	0.08676	0.23370	18	-1
29	12.59054	2	7	0.13995	0.14694	11	3
30	12.73049	2	6	0.16294	0.00699	14	1
31	12.73748	1	7	0.15595	0.29255	19	3
32	12.89343	2	6	1.00487	0.13660	14	1
33	13.03003	1	7	0.86827	0.92548	20	1
34	13.89830	2	6	0.20764	0.05721	12	4
35	13.95551	1	7	0.15043	0.12208	21	4
36	14.07759	1	8	0.02835	2.12163	22	-1
37	14.10594	2	7	3.29817	2.09328	7	5
38	16.19922	1	8	1.20489	2.29762	23	5
39	17.40411	2	7	0.20936	1.09273	15	2
40	17.61347	2	6	0.08997	0.88337	23	5
41	17.70344	2	5	1.32207	0.79340	22	3
42	18.49684	1	6	0.52867	2.43896	24	2
43	19.02551	2	5	0.00820	1.91029	20	1
44	19.03371	2	4	0.12816	1.90209	18	6
45	19.16187	2	3	1.30966	1.77393	24	2
46	20.47153	2	2	-1.00000	0.46427	21	4
47	20.93580	1	3	7.01508	0.96377	25	1
48	21.89957	1	4	0.22643	0.89345	26	2
49	22.12600	2	3	5.82488	0.66702	26	2
50	22.79302	1	4	3.25533	0.03045	27	2
51	22.82347	1	5	1.06988	1.86238	28	3
52	23.89335	2	4	2.15500	0.79250	28	3
53	24.68585	1	5	1.36250	0.76852	29	3
54	25.45437	1	6	0.59398	2.35404	30	4
55	26.04835	2	5	1.90253	1.76006	27	2

56	27.80841	1	6	0.14247	0.44775	31	2
57	27.95088	2	5	0.28838	0.30528	25	1
58	28.23926	2	4	2.92459	0.01690	31	2
59	28.25616	1	5	2.42331	1.17109	32	1
60	29.42725	1	6	1.25222	2.60533	33	2
61	30.67947	2	5	0.48438	1.35311	32	1
62	31.16385	2	4	0.37613	0.86873	30	4
63	31.53998	2	3	0.43527	0.49260	29	3
64	31.97525	2	2	-1.00000	0.05733	33	2
65	32.03258	1	3	2.63538	0.45210	34	1
66	32.48468	1	4	2.18328	0.61496	35	2
67	33.09964	1	5	1.56832	0.79783	36	3
68	33.89747	1	6	0.77049	0.39830	37	4
69	34.29577	1	7	0.08825	3.11383	38	5
70	34.38402	2	6	0.28394	3.02558	38	5
71	34.66796	2	5	0.50813	2.74164	34	1
72	35.17609	2	4	2.69702	2.23351	36	3
73	37.40960	1	5	0.46351	0.62368	39	1
74	37.87311	2	4	0.01090	0.16017	35	2
75	37.88401	2	3	1.35844	0.14927	37	4
76	38.03328	1	4	0.77511	2.75571	40	2
77	38.80839	2	3	0.43406	1.98060	40	2
78	39.24245	2	2	-1.00000	1.54654	39	1
79	40.78899	1	3	6.16869	0.12603	41	1
80	40.91502	1	4	1.26224	1.41287	42	2
81	42.17726	2	3	4.78042	0.15063	42	2
82	42.32789	1	4	1.55046	1.76931	43	2
83	43.87835	2	3	3.07933	0.21885	43	2
84	44.09720	1	4	2.86048	0.55607	44	2
85	44.65327	1	5	0.48324	0.70809	45	3

86	45.13651	2	4	1.82117	0.22485	45	3
87	45.36136	1	5	1.33532	2.10171	46	3
88	46.69668	2	4	0.26100	0.76639	46	3
89	46.95768	2	3	5.27834	0.50539	41	1
90	47.46307	1	4	4.77295	2.46702	47	1
91	49.93009	1	5	2.30593	0.28418	48	3
92	50.21427	1	6	1.02824	2.47504	49	4
93	51.24251	2	5	0.99351	1.44680	49	4
94	52.23602	2	4	0.90826	0.45329	44	2
95	52.68931	1	5	0.45497	0.09644	50	2
96	52.78575	1	6	0.35853	1.14405	51	4
97	53.14428	2	5	3.67950	0.78552	48	3
98	53.92980	1	6	2.79883	1.28056	52	3
99	55.21036	1	7	1.51827	2.09126	53	5
100	56.72863	2	6	0.09515	0.57299	52	3

Таблица 4.2 Данные о заявках

$j$	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$	$k(j)$
1	0.51505	0	0	0.51505	2.16997	2.68502	1
2	2.53474	0	0	2.53474	6.46140	8.99614	2
3	2.71801	0	0	2.71801	0.58301	3.30102	1
4	3.08670	0	0	3.08670	2.85405	5.94075	3
5	3.75735	0	0	3.75735	5.33338	9.09073	1
6	4.62467	0	0	4.62467	4.15817	8.78284	4
7	5.51736	0	0	5.51736	8.58858	14.10594	5
8	6.40146	0	0	6.40146	1.09370	7.49516	3
9	7.30990	0	0	7.30990	0.03471	7.34461	6
10	7.44191	0	0	7.44191	1.28536	8.72727	6
11	7.80294	0	0	7.80294	4.78760	12.59054	3
12	8.51895	1	4.07	12.59054	1.30776	13.89830	4
13	9.65955	0	0	9.65955	0.33144	9.99099	1

14	10.74315	0	0	10.74315	2.15028	12.89343	1
15	11.93680	0	0	11.93680	5.46731	17.40411	2
16	12.12377	0	0	12.12377	-1.00000	-1.00000	4
17	12.39786	1	0	12.39786	-1.00000	-1.00000	6
18	12.50378	2	1.6	14.10594	4.92777	19.03371	6
19	12.73748	1	0	12.73748	-1.00000	-1.00000	3
20	13.03003	1	0	13.03003	5.99548	19.02551	1
21	13.95551	1	0	13.95551	6.51602	20.47153	4
22	14.07759	2	3.33	17.40411	0.29933	17.70344	3
23	16.19922	2	0	16.19922	1.41425	17.61347	5
24	18.49684	0	0	18.49684	0.66503	19.16187	2
25	20.93580	0	0	20.93580	7.01508	27.95088	1
26	21.89957	0	0	21.89957	0.22643	22.12600	2
27	22.79302	0	0	22.79302	3.25533	26.04835	2
28	22.82347	0	0	22.82347	1.06988	23.89335	3
29	24.68585	0	0	24.68585	6.85413	31.53998	3
30	25.45437	0	0	25.45437	5.70948	31.16385	4
31	27.80841	0	0	27.80841	0.43085	28.23926	2
32	28.25616	0	0	28.25616	2.42331	30.67947	1
33	29.42725	0	0	29.42725	2.54800	31.97525	2
34	32.03258	0	0	32.03258	2.63538	34.66796	1
35	32.48468	0	0	32.48468	5.38843	37.87311	2
36	33.09964	0	0	33.09964	2.07645	35.17609	3
37	33.89747	0	0	33.89747	3.98654	37.88401	4
38	34.29577	1	0	34.29577	0.08825	34.38402	5
39	37.40960	0	0	37.40960	1.83285	39.24245	1
40	38.03328	0	0	38.03328	0.77511	38.80839	2
41	40.78899	0	0	40.78899	6.16869	46.95768	1
42	40.91502	0	0	40.91502	1.26224	42.17726	2
43	42.32789	0	0	42.32789	1.55046	43.87835	2

44	44.09720	0	0	44.09720	8.13882	52.23602	2
45	44.65327	0	0	44.65327	0.48324	45.13651	3
46	45.36136	0	0	45.36136	1.33532	46.69668	3
47	47.46307	0	0	47.46307	-1.00000	-1.00000	1
48	49.93009	0	0	49.93009	3.21419	53.14428	3
49	50.21427	0	0	50.21427	1.02824	51.24251	4
50	52.68931	0	0	52.68931	-1.00000	-1.00000	2
51	52.78575	0	0	52.78575	-1.00000	-1.00000	4
52	53.92980	0	0	53.92980	2.79883	56.72863	3
53	55.21036	1	0	55.21036	-1.00000	-1.00000	5

Таблица 4.3 Данные о приборах

$k$	$N(k)$	$t_{зан}(k)$	$t_{np}(k)$	$\Delta_{np}(k)$
1	12	56.05526	0.67337	0.01187
2	13	40.92671	15.80192	0.27855
3	12	33.20523	23.52340	0.41467
4	8	27.35096	29.37767	0.51786
5	4	21.33835	35.39028	0.62385
6	4	19.61304	37.11559	0.65427
	53	198.48955	141.88223	2.50107

число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ :

**53**

число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ :

**47**

среднее число заявок, находившихся в СМО, на интервале  $[0, t_{cob}(100)]$ :

**4.7**

среднее время пребывания заявок в очереди на интервале  $[0, t_{cob}(100)]$ :

**0.1915**

среднее время пребывания заявок в СМО на интервале  $[0, t_{cob}(100)]$ :

**1.10546**

## Задание 5

Система массового обслуживания (M|M|n|m)

$V=90$ ,  $\lambda=1.142$ ,  $\mu_2=0.235$ ,  $n=6$ ,  $m=13$

Таблица 5.1 Данные о событиях

$l$	$t_{\text{соб}}(l)$	$Type(l)$	$C(l)$	$t_{\text{осмин}}(l)$	$t_{\text{ож}}(l)$	$j(l)$	$k(l)$
1	0.45652	1	1	3.83412	0.51058	1	1
2	0.96710	1	2	3.32354	1.19519	2	2
3	2.16229	1	3	0.14666	0.07012	3	3
4	2.23241	1	4	0.07654	0.58396	4	4
5	2.30895	2	3	1.98169	0.50742	3	3
6	2.81637	1	4	1.47427	0.46587	5	3
7	3.28224	1	5	1.00840	1.51019	6	5
8	4.29064	2	4	0.50786	0.50179	1	1
9	4.79243	1	5	0.00149	0.76121	7	1
10	4.79392	2	4	0.00458	0.75972	7	1
11	4.79850	2	3	3.41241	0.75514	2	2
12	5.55364	1	4	2.65727	0.15198	8	1
13	5.70562	1	5	1.75233	0.67995	9	2
14	6.38557	1	6	1.07238	0.33083	10	6
15	6.71640	1	7	0.74155	0.18319	11	-1
16	6.89959	1	8	0.55836	0.41246	12	-1
17	7.31205	1	9	0.14590	0.24375	13	-1
18	7.45795	2	8	0.75296	0.09785	9	2
19	7.55580	1	9	0.65511	0.25310	14	2
20	7.80890	1	10	0.40201	1.12049	15	-1
21	8.21091	2	9	0.32336	0.71848	6	5
22	8.53427	2	8	0.00790	0.39512	14	2
23	8.54217	2	7	0.18101	0.38722	8	1
24	8.72318	2	6	3.12746	0.20621	12	6
25	8.92939	1	7	1.07454	0.62307	16	1

26	9.55246	1	8	0.45147	0.12083	17	5
27	9.67329	1	9	0.33064	3.26066	18	6
28	10.00393	2	8	1.84671	2.93002	16	1
29	11.85064	2	7	4.54140	1.08331	4	4
30	12.93395	1	8	0.24587	1.15437	19	1
31	13.17982	2	7	3.21222	0.90850	19	1
32	14.08832	1	8	2.19670	0.64826	20	1
33	14.73658	1	9	1.54844	0.44790	21	4
34	15.18448	1	10	1.10054	0.17480	22	-1
35	15.35928	1	11	0.92574	1.13486	23	-1
36	16.28502	2	10	0.09324	0.20912	20	1
37	16.37826	2	9	0.01378	0.11588	21	4
38	16.39204	2	8	0.47265	0.10210	13	3
39	16.49414	1	9	0.37055	0.60797	24	1
40	16.86469	2	8	2.00164	0.23742	22	2
41	17.10211	1	9	0.98608	0.11670	25	2
42	17.21881	1	10	0.86938	0.01545	26	3
43	17.23426	1	11	0.85393	0.58370	27	4
44	17.81796	1	12	0.27023	0.53362	28	-1
45	18.08819	2	11	0.12223	0.26339	25	2
46	18.21042	2	10	0.65591	0.14116	27	4
47	18.35158	1	11	0.51475	1.95885	29	2
48	18.86633	2	10	2.84326	1.44410	18	6
49	20.31043	1	11	1.39916	0.50590	30	4
50	20.81633	1	12	0.89326	2.12118	31	6
51	21.70959	2	11	0.19307	1.22792	24	1
52	21.90266	2	10	0.68703	1.03485	23	5
53	22.58969	2	9	1.07741	0.34782	31	6
54	22.93751	1	10	0.72959	0.65706	32	1
55	23.59457	1	11	0.07253	0.67956	33	5



56	23.66710	2	10	0.31492	0.60703	29	2
57	23.98202	2	9	0.49089	0.29211	30	4
58	24.27413	1	10	0.19878	3.80490	34	2
59	24.47291	2	9	1.07617	3.60612	32	1
60	25.54908	2	8	0.05846	2.52995	34	2
61	25.60754	2	7	2.12093	2.47149	33	5
62	27.72847	2	6	-1.00000	0.35056	28	3
63	28.07903	1	7	6.91251	6.55270	35	1
64	34.63173	1	8	0.35981	0.33472	36	2
65	34.96645	1	9	0.02509	0.82477	37	3
66	34.99154	2	8	1.43617	0.79968	35	1
67	35.79122	1	9	0.63649	0.00459	38	1
68	35.79581	1	10	0.63190	0.17312	39	4
69	35.96893	1	11	0.45878	1.29147	40	5
70	36.42771	2	10	0.36623	0.83269	36	2
71	36.79394	2	9	0.08549	0.46646	39	4
72	36.87943	2	8	0.41198	0.38097	37	3
73	37.26040	1	9	0.03101	0.05221	41	2
74	37.29141	2	8	1.51163	0.02120	38	1
75	37.31261	1	9	1.49043	0.32259	42	1
76	37.63520	1	10	1.16784	2.50338	43	3
77	38.80304	2	9	1.15502	1.33554	40	5
78	39.95806	2	8	5.31910	0.18052	43	3
79	40.13858	1	9	2.22794	0.28434	44	3
80	40.42292	1	10	1.94360	0.81923	45	4
81	41.24215	1	11	0.19283	1.62654	46	5
82	41.43498	2	10	0.93154	1.43371	46	5
83	42.36652	2	9	0.14666	0.50217	44	3
84	42.51318	2	8	2.76398	0.35551	45	4
85	42.86869	1	9	2.40847	0.12054	47	3

86	42.98923	1	10	2.28793	0.31709	48	4
87	43.30632	1	11	1.97084	0.75922	49	5
88	44.06554	1	12	0.15921	0.86051	50	6
89	44.22475	2	11	1.05241	0.70130	50	6
90	44.92605	1	12	0.35111	2.32828	51	6
91	45.27716	2	11	0.19111	1.97717	42	1
92	45.46827	2	10	0.73236	1.78606	41	2
93	46.20063	2	9	1.48186	1.05370	47	3
94	47.25433	1	10	0.42816	0.44510	52	1
95	47.68249	2	9	4.65397	0.01694	48	4
96	47.69943	1	10	4.63703	1.13655	53	2
97	48.83598	1	11	0.35088	2.10985	54	3
98	49.18686	2	10	3.14960	1.75897	54	3
99	50.94583	1	11	0.86028	2.08622	55	3
100	51.80611	2	10	0.53035	1.22594	55	3

Таблица 5.2 Данные о заявках

$j$	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}(j)$	$t_{обсл}(j)$	$t_{коб}(j)$	$k(j)$
1	0.45652	0	0.00000	0.45652	3.83412	4.29064	1
2	0.96710	0	0.00000	0.96710	3.83140	4.79850	2
3	2.16229	0	0.00000	2.16229	0.14666	2.30895	3
4	2.23241	0	0.00000	2.23241	9.61823	11.85064	4
5	2.81637	0	0.00000	2.81637	-1.00000	-1.00000	3
6	3.28224	0	0.00000	3.28224	4.92867	8.21091	5
7	4.79243	0	0.00000	4.79243	0.00149	4.79392	1
8	5.55364	0	0.00000	5.55364	2.98853	8.54217	1
9	5.70562	0	0.00000	5.70562	1.75233	7.45795	2
10	6.38557	0	0.00000	6.38557	-1.00000	-1.00000	6
11	6.71640	1	0.74155	7.45795	-0.74155	6.71640	3
12	6.89959	2	1.31132	8.21091	0.51227	8.72318	6
13	7.31205	3	1.22222	8.53427	7.85777	16.39204	3

14	7.55580	3	0.00000	7.55580	0.97847	8.53427	2
15	7.80890	4	0.73327	8.54217	-0.73327	7.80890	2
16	8.92939	1	0.00000	8.92939	1.07454	10.00393	1
17	9.55246	2	0.00000	9.55246	-1.00000	-1.00000	5
18	9.67329	3	0.00000	9.67329	9.19304	18.86633	6
19	12.93395	2	0.00000	12.93395	0.24587	13.17982	1
20	14.08832	2	0.00000	14.08832	2.19670	16.28502	1
21	14.73658	3	0.00000	14.73658	1.64168	16.37826	4
22	15.18448	4	1.10054	16.28502	0.57967	16.86469	2
23	15.35928	5	1.01898	16.37826	5.52440	21.90266	5
24	16.49414	3	0.00000	16.49414	5.21545	21.70959	1
25	17.10211	3	0.00000	17.10211	0.98608	18.08819	2
26	17.21881	4	0.00000	17.21881	-1.00000	-1.00000	3
27	17.23426	5	0.00000	17.23426	0.97616	18.21042	4
28	17.81796	6	0.27023	18.08819	9.64028	27.72847	3
29	18.35158	5	0.00000	18.35158	5.31552	23.66710	2
30	20.31043	5	0.00000	20.31043	3.67159	23.98202	4
31	20.81633	6	0.00000	20.81633	1.77336	22.58969	6
32	22.93751	4	0.00000	22.93751	1.53540	24.47291	1
33	23.59457	5	0.00000	23.59457	2.01297	25.60754	5
34	24.27413	4	0.00000	24.27413	1.27495	25.54908	2
35	28.07903	1	0.00000	28.07903	6.91251	34.99154	1
36	34.63173	2	0.00000	34.63173	1.79598	36.42771	2
37	34.96645	3	0.00000	34.96645	1.91298	36.87943	3
38	35.79122	3	0.00000	35.79122	1.50019	37.29141	1
39	35.79581	4	0.00000	35.79581	0.99813	36.79394	4
40	35.96893	5	0.00000	35.96893	2.83411	38.80304	5
41	37.26040	3	0.00000	37.26040	8.20787	45.46827	2
42	37.31261	3	0.00000	37.31261	7.96455	45.27716	1
43	37.63520	4	0.00000	37.63520	2.32286	39.95806	3

44	40.13858	3	0.00000	40.13858	2.22794	42.36652	3
45	40.42292	4	0.00000	40.42292	2.09026	42.51318	4
46	41.24215	5	0.00000	41.24215	0.19283	41.43498	5
47	42.86869	3	0.00000	42.86869	3.33194	46.20063	3
48	42.98923	4	0.00000	42.98923	4.69326	47.68249	4
49	43.30632	5	0.00000	43.30632	-1.00000	-1.00000	5
50	44.06554	6	0.00000	44.06554	0.15921	44.22475	6
51	44.92605	6	0.00000	44.92605	-1.00000	-1.00000	6
52	47.25433	4	0.00000	47.25433	-1.00000	-1.00000	1
53	47.69943	4	0.00000	47.69943	-1.00000	-1.00000	2
54	48.83598	5	0.00000	48.83598	0.35088	49.18686	3
55	50.94583	5	0.00000	50.94583	0.86028	51.80611	3

Таблица 5.3

$k$	$N(k)$	$t_{зан}(k)$	$t_{np}(k)$	$\Delta_{np}(k)$
1	12	38.04090	13.76521	0.26571
2	11	44.14222	7.66389	0.14793
3	12	47.50092	4.30519	0.08310
4	7	23.68931	28.11680	0.54273
5	7	41.48872	10.31739	0.19915
6	6	51.80611	0.00000	0.00000
	55	41.11000	10.69000	0.21000

число заявок  $J(100)$ , поступивших в СМО на интервале  $[0, t_{cob}(100)]$ :

**55**

число  $JF(100)$  полностью обслуженных заявок на интервале  $[0, t_{cob}(100)]$ :

**45**

число  $JL(100)$  отклоненных заявок на интервале  $[0, t_{cob}(100)]$ :

**0**

среднее число заявок, находившихся в СМО, на интервале  $[0, t_{cob}(100)]$ :

**8.52**

среднее время пребывания заявок в очереди на интервале  $[0, t_{cob}(100)]$ :

**0.14218**

среднее время пребывания заявок в СМО на интервале  $[0, t_{\text{cob}}(100)]$ :

**0.93025**

### **Список литературы**

1. Лобузов А.А. Системы массового обслуживания [Электронный ресурс]: методические указания. – М.: РТУ МИРЭА, 2022.
2. Лобузов А.А., Гумляева С.Д., Норин Н.В. Задачи по теории случайных процессов. – М.: МИРЭА, 1993. – 68 с.
3. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. – М.: ЛКИ, 2021. – 400 с.
4. Кирпичников А.П. Методы прикладной теории массового обслуживания. – М.: URSS, 2018. – 224 с.
5. Ивченко Г.И., Каштанов В.А., Коваленко И.Н. Теория массового обслуживания. – М.: URSS, 2012. – 304 с.

## Приложение

### Приложение 1. Код задачи 1

```
n = 6
m = 13
deltaT_z = 0.724 #(1)
T_ob = 0.738
lamda = 1.142
mu_1 = 1.412 #(1)
mu_2 = 0.235

import numpy as np

def rfind(arr, item):
    b = list(reversed(arr))
    return len(arr) - b.index(item) - 1

t_ob = round(np.random.exponential(1/mu_1), 5)

l = [1]
t_sob = [deltaT_z]
Type = [1]
C_l = [1]
t_ost = [t_ob]
t_ozh = [deltaT_z]
j_l = [1]

for i in range(1, 100):
    l.append(i + 1)
    if (Type[i-1] == 1 or Type[i-1] == 3):
        if(t_sob[i-1] + t_ost[i-1] < t_sob[i-1] + t_ozh[i-1]):
            t_sob.append(round(t_sob[i-1] + t_ost[i-1], 5))
            Type.append(2)
            C_l.append(0)
            t_ost.append(-1)
            t_ozh.append(round(t_ozh[i-1] - t_ost[i-1], 5))
            if (Type[i-1] == 3):
                j_l.append(j_l[rfind(Type, 1)])
            else:
                j_l.append(j_l[i-1])
        else:
            pass
```

```

    t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
    Type.append(3)
    C_l.append(1)
    t_ost.append(round(t_ost[i-1] - t_ozh[i-1], 5))
    t_ozh.append(deltaT_z)
    j_l.append(j_l[i-1] + 1)
elif(Type[i-1] == 2):
    t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
    Type.append(1)
    C_l.append(1)
    t_ost.append(round(np.random.exponential(1/mu_1), 5))
    t_ozh.append(deltaT_z)
    j_l.append(max(j_l) + 1)

print(f"!t_sob;Type;C_l;t_ost;t_ozh;j_l")
for i in range(100):
    print(f"!{i};{t_sob[i]};{Type[i]};{C_l[i]};{t_ost[i]};{t_ozh[i]};{j_l[i]}")

t_z = []
t_obsl = []
t_kob = []

for i in range(1, max(j_l) + 1):
    t_z.append(t_sob[j_l.index(i)])
    if (Type[j_l.index(i)] != 3):
        t_obsl.append(t_ost[j_l.index(i)])
        t_kob.append(t_z[i-1] + t_obsl[i-1])
    else:
        t_obsl.append(0)
        t_kob.append(t_z[i-1])

#print(len(t_z))
#print("t_z: ", t_z)
#print("t_obsl", t_obsl)
#print("t_kob: ", t_kob)

print("j;t_z;t_obsl;t_kob")
for i in range(max(j_l)):
    print(f"{i};{t_z[i]};{t_obsl[i]};{t_kob[i]}")

R_0 = C_l.count(0)

```



```

R_1 = C_l.count(1)

v_0 = R_0 / 100
v_1 = R_1 / 100

T_0 = 0
T_1 = 0

for i in range(100):
    if (C_l[i] == 1):
        T_1 += t_ost[i]
T_0 = max(t_sob) - T_1

T_0 = round(T_0, 5)
T_1 = round(T_1, 5)

delta_0 = round(T_0 / max(t_sob), 5)
delta_1 = round(T_1 / max(t_sob), 5)

print("R_i v_i T_i D_i")
print(R_0, v_0, T_0, delta_0)
print(R_1, v_1, T_1, delta_1)
print(R_0 + R_1, v_0 + v_1, T_0 + T_1, delta_0 + delta_1)

#print(R_0, R_1)
#print(v_0, v_1)
#print(T_0, T_1)
#print(delta_0, delta_1)

J = max(j_l)
JF = 0
for i in range(100):
    if (Type[i] == 2):
        JF += 1
JL = J - JF
T_zan = T_0
T_svob = T_1

print(f"J: {J}")
print(f"JF: {JF}")
print(f"JL: {JL}")

```

```
print(f"T_zan: {T_zan}")
print(f"T_svob: {T_svob}")
```

## Приложение 2. Код задачи 2

```
n = 6
m = 13
deltaT_z = 0.724
T_ob = 0.738 #(2)
lamda = 1.142 #(2)
mu_1 = 1.412
mu_2 = 0.235

import numpy as np

def rfind(arr, item):
    b = list(reversed(arr))
    return len(arr) - b.index(item) - 1

t_z = round(np.random.exponential(1/lamda), 5)

l = [1]
t_sob = [t_z]
Type = [1]
C_l = [1]
t_ost = [T_ob]
t_ozh = [round(np.random.exponential(1/mu_1), 5)]
j_l = [1]

for i in range(1, 100):
    l.append(i + 1)
    if(t_ost[i-1] < t_ozh[i-1] and t_ost[i-1] != -1):
        t_sob.append(round(t_sob[i-1] + T_ob, 5))
        Type.append(2)
        C_l.append(C_l[i-1] - 1)
        if(C_l[i] != 0):
            t_ost.append(round(T_ob, 5))
        else:
            t_ost.append(-1)
        t_ozh.append(round(t_ozh[i-1] - t_ost[i-1], 5))
        j_l.append(max(j_l) - C_l[i])
```

```

else:
    t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
    Type.append(1)
    C_l.append(C_l[i-1] + 1)
    if (t_ost[i-1] != -1):
        t_ost.append(round(t_ost[i-1] - t_ozh[i-1], 5))
    else:
        t_ost.append(T_ob)
    t_ozh.append(round(np.random.exponential(1/mu_1), 5))
    j_l.append(max(j_l) + 1)

print(l)
print(t_sob)
print(Type)
print(C_l)
print(t_ost)
print(t_ozh)
print(j_l)

print("l;t_sob;Type;C_l;t_ost;t_ozh;j_l")
for i in range(100):
    print(f" {l[i]};{t_sob[i]};{Type[i]};{C_l[i]};{t_ost[i]};{t_ozh[i]};{j_l[i]}")

t_z = []
q_j = []
t_och = []
t_nob = []
t_obsl = []
t_kob = []
j = []

for i in range(1, max(j_l) + 1):
    j.append(i)
    t_z.append(t_sob[j_l.index(i)])

    if(C_l[j_l.index(i)] > 1):
        q_j.append(C_l[j_l.index(i)] - 1)
        t_och.append(round(t_ost[j_l.index(i)] + T_ob * q_j[i-1], 5))
        t_nob.append(round(t_z[i-1] + t_och[i-1], 5))
    else:
        q_j.append(0)

```

```

t_och.append(0)
t_nob.append(t_z[i-1])

t_obs1.append(T_ob)
t_kob.append(round(t_nob[i-1] + T_ob, 5))

print("j;t_z;q_j;t_och;t_nob;t_obs1;t_kob")
for i in range(len(j)):
    print(f"{j[i]};{t_z[i]};{q_j[i]};{t_och[i]};{t_nob[i]};{t_obs1[i]};{t_kob[i]}")

R = []
v = []
T = list(np.zeros(len(set(C_l))))
delta = []

for i in range(0, max(C_l) + 1):
    R.append(C_l.count(i))
    v.append(R[i] / 100)
    for j in range(99):
        if (C_l[j] == i):
            T[i] += round(t_sob[j+1] - t_sob[j], 5)
    T[i] = round(T[i], 5)
    delta.append(round(T[i] / max(t_z), 5))

print("R;v;T;delta")
for i in range(len(R)):
    print(f"{R[i]};{v[i]};{T[i]};{delta[i]}")

#print(R)
#print(v)
#print(T)
#print(delta)

for i in range(len(R)):
    print(delta[i])

J = max(j_l)
JF = 0
for i in range(100):
    if (Type[i] == 2):
        JF += 1

```

```

z = round(sum(C_l) / 100, 5)
_t_och = round(sum(t_och) / JF, 5)
_t_smo = round((sum(t_kob) - sum(t_z)) / JF, 5)

T_svob = round(T[0], 5)

print("J:", J)
print("JF: ", JF)
print("z: ", z)
print("_t_och: ", _t_och)
print("_t_smo: ", _t_smo)
print("T_svob: ", T_svob)

```

### Приложение 3. Код задачи 3

```

n = 6
m = 13
deltaT_z = 0.724
T_ob = 0.738
lamda = 1.142 #(3)
mu_1 = 1.412 #(3)
mu_2 = 0.235

import numpy as np

def rfind(arr, item):
    b = list(reversed(arr))
    return len(arr) - b.index(item) - 1

t_z = round(np.random.exponential(1/lamda), 5)
t_ob = round(np.random.exponential(1/mu_1), 5)

l = [1]
t_sob = [t_z]
Type = [1]
C_l = [1]
t_ost = [t_ob]
t_ozh = [round(np.random.exponential(1/lamda), 5)]
j_l = [1]

for i in range(1, 100):

```

```

l.append(i + 1)
if(t_ost[i-1] < t_ozh[i-1] and t_ost[i-1] != -1):
    t_sob.append(round(t_sob[i-1] + t_ost[i-1], 5))
    Type.append(2)
    C_l.append(C_l[i-1] - 1)
    if(C_l[i] != 0):
        t_ost.append(round(np.random.exponential(1/mu_1), 5))
    else:
        t_ost.append(-1)
    t_ozh.append(round(t_ozh[i-1] - t_ost[i-1], 5))
    j_l.append(max(j_l) - C_l[i])
else:
    t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
    Type.append(1)
    C_l.append(C_l[i-1] + 1)
    if (t_ost[i-1] != -1):
        t_ost.append(round(t_ost[i-1] - t_ozh[i-1], 5))
    else:
        t_ost.append(round(np.random.exponential(1/mu_1), 5))
    t_ozh.append(round(np.random.exponential(1/lamda), 5))
    j_l.append(max(j_l) + 1)

```

```

#print(l)
#print(t_sob)
#print(Type)
#print(C_l)
#print(t_ost)
#print(t_ozh)
#print(j_l)

```

```

print("l;t_sob;Type;C_l;t_ost;t_ozh;j_l")
for i in range(100):
    print(f" {l[i]};{t_sob[i]};{Type[i]};{C_l[i]};{t_ost[i]};{t_ozh[i]};{j_l[i]}")

```

```

j = []
t_z = []
q_j = []
t_och = []
t_nob = []
t_obsl = []
t_kob = []

```

```

for i in range(1, max(j_l) + 1):
    j.append(i)
    t_z.append(t_sob[j_l.index(i)])

    if(C_l[j_l.index(i)] > 1):
        q_j.append(C_l[j_l.index(i)] - 1)
        t_och.append(round(t_sob[rfind(j_l, i-1)] - t_z[i-1], 5))
        if (t_och[i-1] < 0):
            t_och[i-1] = -1
            t_nob.append(-1)
            t_obsl.append(-1)
            t_kob.append(-1)
        else:
            t_nob.append(round(t_z[i-1] + t_och[i-1], 5))
            t_obsl.append(round(t_sob[rfind(j_l, i)] - t_sob[rfind(j_l, i-1)], 5))
            t_kob.append(round(t_nob[i-1] + t_obsl[i-1], 5))
    else:
        q_j.append(0)
        t_och.append(0)
        t_nob.append(t_z[i-1])
        t_obsl.append(t_ost[j_l.index(i)])
        t_kob.append(round(t_nob[i-1] + t_obsl[i-1], 5))

print(j)
print(t_z)
print(q_j)
print(t_och)
print(t_nob)
print(t_obsl)
print(t_kob)

print("j;t_z;q_j;t_och;t_nob;t_obsl;t_kob")
for i in range(len(j)):
    print(f"{j[i]};{t_z[i]};{q_j[i]};{t_och[i]};{t_nob[i]};{t_obsl[i]};{t_kob[i]}")

R = []
v = []
T = list(np.zeros(len(set(C_l))))
delta = []

```

```

for i in range(0, max(C_l) + 1):
    R.append(C_l.count(i))
    v.append(R[i] / 100)
    for j in range(99):
        if (C_l[j] == i):
            T[i] += t_sob[j+1] - t_sob[j]
    T[i] = round(T[i], 5)
    delta.append(round(T[i] / max(t_z), 5))
print(sum(T))

print("R;v;T;delta")
for i in range(len(R)):
    print(f"{R[i]};{v[i]};{T[i]};{delta[i]}")

J = max(j_l)
JF = 0
for i in range(100):
    if (Type[i] == 2):
        JF += 1
z = round(sum(C_l) / 100, 5)
_t_och = round(sum(t_och) / JF, 5)
_t_smo = 0
for i in range(JF):
    _t_smo += round(t_kob[i] - t_z[i], 5)

_t_smo = _t_smo / 100
T_svob = round(T[0], 5)

print("J:", J)
print("JF: ", JF)
print("z: ", z)
print("_t_och: ", _t_och)
print("_t_smo: ", _t_smo)
print("T_svob: ", T_svob)

```

#### Приложение 4. Код задачи 4

```

n = 6 #(4)
m = 13
deltaT_z = 0.724
T_ob = 0.738

```



```

lamda = 1.142 #(4)
mu_1 = 1.412
mu_2 = 0.235 #(4)

import numpy as np

def rfind(arr, item):
    b = list(reversed(arr))
    return len(arr) - b.index(item) - 1

def min_not_null(arr):
    m = 9999999
    for i in range(len(arr)):
        if(arr[i] < m)and(arr[i] != 0):
            m = arr[i]
    return m

t_z = round(np.random.exponential(1/lamda), 5)
t_ob = round(np.random.exponential(1/mu_2), 5)
och = []

prib_time = []
prib_z = []
for i in range(100):
    prib_time.append([])
    prib_z.append([])
    for j in range(n):
        prib_time[i].append(0)
        prib_z[i].append(0)

prib_time[0][0] = t_ob
prib_z[0][0] = 1

l = [1]
t_sob = [t_z]
Type = [1]
C_l = [1]
t_osmin = [t_ob]
t_ozh = [round(np.random.exponential(1/lamda), 5)]
j_l = [1]
k_l = [1]

```

```

for i in range(1, 100):
    if och != []:
        print(och)
    l.append(i + 1)
    if(t_osmin[i-1] < t_ozh[i-1] and t_osmin[i-1] != -1): # Завершение обработки
        t_sob.append(round(t_sob[i-1] + t_osmin[i-1], 5))
        delta_t = t_sob[i] - t_sob[i-1]
        Type.append(2)
        C_l.append(C_l[i-1] - 1)
        for j in range(n):
            if(prib_time[i-1][j] != 0):
                prib_time[i][j] = round(prib_time[i-1][j] - delta_t, 5)
                prib_z[i][j] = prib_z[i-1][j]
        if(C_l[i] != 0):
            t_osmin.append(min_not_null(prib_time[i]))
        else:
            t_osmin.append(-1)
        t_ozh.append(round(t_ozh[i-1] - t_osmin[i-1], 5))
        j_l.append(prib_z[i][prib_time[i-1].index(min_not_null(prib_time[i-1]))])
        prib_z[i][prib_time[i-1].index(min(prib_time[i-1]))] = 0
        k_l.append(prib_time[i-1].index(min_not_null(prib_time[i-1])) + 1)
        if(C_l[i] > n):
            prib_time[i][k_l[i]] = round(np.random.exponential(1/mu_2), 5)
            prib_z[i][k_l[i]] = och.pop(0)

    else: # Появление новой заявки
        t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
        delta_t = t_sob[i] - t_sob[i-1]
        Type.append(1)
        C_l.append(C_l[i-1] + 1)
        for j in range(n):
            if(prib_time[i-1][j] != 0):
                prib_time[i][j] = round(prib_time[i-1][j] - delta_t, 5)
                prib_z[i][j] = prib_z[i-1][j]
        if (t_osmin[i-1] != -1):
            if(min(prib_time[i]) == 0):
                k_l.append(prib_time[i].index(0) + 1)
                prib_time[i][k_l[i]-1] = round(np.random.exponential(1/mu_2), 5)
                prib_z[i][k_l[i]-1] = max(j_l) + 1
            else:

```

```

    k_l.append(-1)
    och.append(max(j_l) + 1)
    t_osmin.append(min_not_null(prib_time[i]))
else:
    t_osmin.append(round(np.random.exponential(1/mu_2), 5))
    prib_time[i][0] = round(t_osmin[i], 5)
    prib_z[i][0] = 1
    k_l.append(1)
    t_ozh.append(round(np.random.exponential(1/lamda), 5))
    j_l.append(max(j_l) + 1)

print("l;t_sob;Type;C_l;t_osmin;t_ozh;j_l;k_l")
for i in range(100):
    print(f"{l[i]};{t_sob[i]};{Type[i]};{C_l[i]};{t_osmin[i]};{t_ozh[i]};{j_l[i]};{k_l[i]}")

j = []
t_z = []
q_j = []
t_och = []
t_nob = []
t_obsl = []
t_kob = []
k_j = []

for i in range(1, max(j_l) + 1):
    j.append(i)
    t_z.append(t_sob[j_l.index(i)])

for k in range(100):
    if(i in prib_z[k]):
        t_nob.append(t_sob[k])
        t_och.append(round(t_nob[i-1] - t_z[i-1], 5))
        k_j.append(prib_z[k].index(i) + 1)
        t_kob.append(t_sob[rfind(j_l, i)])
        if(t_kob[i-1] == t_nob[i-1]):
            t_kob[i-1] = -1
            t_obsl.append(-1)
        else:
            t_obsl.append(round(t_kob[i-1] - t_nob[i-1], 5))
        break

```

```

if(C_l[j_l.index(i)] > n):
    q_j.append(C_l[j_l.index(i)] - n)
else:
    q_j.append(0)

#print(j)
#print(t_z)
#print(q_j)
#print(t_och)
#print(t_nob)
#print(t_obs1)
#print(t_kob)
#print(k_j)

print("j;t_z;q_j;t_och;t_nob;t_obs1;t_kob;k_j")
for i in range(len(j)):
    print(f"{j[i]};{t_z[i]};{q_j[i]};{t_och[i]};{t_nob[i]};{t_obs1[i]};{t_kob[i]};{k_j[i]}")

k = []
N_k = []
t_zan = [0 for x in range(n)]
t_pr = []
delta_pr = []

for i in range(n):
    k.append(i+1)
    tmp = set()
    for t in range(100):
        if(prib_z[t][i] != 0):
            tmp.add(prib_z[t][i])
    N_k.append(len(tmp))
    tmp = set()
    for t in range(1, 100):
        if(prib_z[t][i] not in tmp):
            t_zan[i] += round(prib_time[t][i], 5)
            tmp.add(prib_z[t][i])
    t_zan[i] = round(t_zan[i], 5)
    if(t_zan[i] > max(t_sob)):
        t_zan[i] = round(max(t_sob), 5)
    t_pr.append(round(max(t_sob) - t_zan[i], 5))
    delta_pr.append(round(t_pr[i] / max(t_sob), 5))

```

```

#print(k)
#print(N_k)
#print(t_zan)
#print(t_pr)
#print(delta_pr)

print("k;N_k;t_zan;t_pr;delta_pr")
for i in range(len(k)):
    print(f"{k[i]};{N_k[i]};{t_zan[i]};{t_pr[i]};{delta_pr[i]}")

print(sum(N_k))

J = max(j_l)
JF = Type.count(2)
JL = Type.count(3)

z = round(sum(C_l) / 100, 5)
_t_och = round(sum(t_och) / JF, 5)
_t_smo = 0
for i in range(JF):
    _t_smo += round(t_kob[i] - t_z[i], 5)

_t_smo = _t_smo / JF

print("J:", J)
print("JF: ", JF)
print("z: ", z)
print("_t_och: ", _t_och)
print("_t_smo: ", _t_smo)

```

#### Приложение 5. Код задачи 5

```

n = 6 #(5)
m = 13 #(5)
deltaT_z = 0.724
T_ob = 0.738
lamda = 1.142 #(5)
mu_1 = 1.412
mu_2 = 0.235 #(5)

import numpy as np

```

```

def rfind(arr, item):
    b = list(reversed(arr))
    return len(arr) - b.index(item) - 1

def min_not_null(arr):
    m = 9999999
    for i in range(len(arr)):
        if(arr[i] < m)and(arr[i] != 0):
            m = arr[i]
    return m

t_z = round(np.random.exponential(1/lamda), 5)
t_ob = round(np.random.exponential(1/mu_2), 5)
och = []

prib_time = []
prib_z = []
for i in range(100):
    prib_time.append([])
    prib_z.append([])
    for j in range(n):
        prib_time[i].append(0)
        prib_z[i].append(0)

prib_time[0][0] = t_ob
prib_z[0][0] = 1

l = [1]
t_sob = [t_z]
Type = [1]
C_l = [1]
t_osmin = [t_ob]
t_ozh = [round(np.random.exponential(1/lamda), 5)]
j_l = [1]
k_l = [1]

for i in range(1, 100):
    if och != []:
        print(och)
    l.append(i + 1)

```

```

if(t_osmin[i-1] < t_ozh[i-1] and t_osmin[i-1] != -1): # Завершение обработки
    t_sob.append(round(t_sob[i-1] + t_osmin[i-1], 5))
    delta_t = t_sob[i] - t_sob[i-1]
    Type.append(2)
    C_l.append(C_l[i-1] - 1)
    for j in range(n):
        if(prib_time[i-1][j] != 0):
            prib_time[i][j] = round(prib_time[i-1][j] - delta_t, 5)
            prib_z[i][j] = prib_z[i-1][j]
    if(C_l[i] != 0):
        t_osmin.append(min_not_null(prib_time[i]))
    else:
        t_osmin.append(-1)
    t_ozh.append(round(t_ozh[i-1] - t_osmin[i-1], 5))
    j_l.append(prib_z[i][prib_time[i-1].index(min_not_null(prib_time[i-1]))])
    prib_z[i][prib_time[i-1].index(min(prib_time[i-1]))] = 0
    k_l.append(prib_time[i-1].index(min_not_null(prib_time[i-1])) + 1)
    if(och != []):
        prib_time[i][k_l[i]] = round(np.random.exponential(1/mu_2), 5)
        prib_z[i][k_l[i]] = och.pop(0)

else: # Появление новой заявки
    t_sob.append(round(t_sob[i-1] + t_ozh[i-1], 5))
    delta_t = t_sob[i] - t_sob[i-1]
    for j in range(n):
        if(prib_time[i-1][j] != 0):
            prib_time[i][j] = round(prib_time[i-1][j] - delta_t, 5)
            prib_z[i][j] = prib_z[i-1][j]
    if(len(och) == m):
        Type.append(2)
        C_l.append(C_l[i-1])
        t_osmin.append(min_not_null(prib_time[i]))
        k_l.append(-1)
    else:
        Type.append(1)
        C_l.append(C_l[i-1] + 1)
        if (t_osmin[i-1] != -1):
            if(min(prib_time[i]) == 0):
                k_l.append(prib_time[i].index(0) + 1)
                prib_time[i][k_l[i]-1] = round(np.random.exponential(1/mu_2), 5)
                prib_z[i][k_l[i]-1] = max(j_l) + 1

```

```

    else:
        k_l.append(-1)
        och.append(max(j_l) + 1)
        t_osmin.append(min_not_null(prib_time[i]))
    else:
        t_osmin.append(round(np.random.exponential(1/mu_2), 5))
        prib_time[i][0] = round(t_osmin[i], 5)
        prib_z[i][0] = 1
        k_l.append(1)
    t_ozh.append(round(np.random.exponential(1/lamda), 5))
    j_l.append(max(j_l) + 1)

#print(l)
#print(t_sob)
#print(Type)
#print(C_l)
#print(t_osmin)
#print(t_ozh)
#print(j_l)
#print(k_l)

print("l;t_sob;Type;C_l;t_osmin;t_ozh;j_l;k_l")
for i in range(100):
    print(f"{l[i]};{t_sob[i]};{Type[i]};{C_l[i]};{t_osmin[i]};{t_ozh[i]};{j_l[i]};{k_l[i]}")

j = []
t_z = []
q_j = []
t_och = []
t_nob = []
t_obsl = []
t_kob = []
k_j = []

for i in range(1, max(j_l) + 1):
    j.append(i)
    t_z.append(t_sob[j_l.index(i)])

for k in range(100):
    if(i in prib_z[k]):
        t_nob.append(t_sob[k])

```



```

    t_och.append(round(t_nob[i-1] - t_z[i-1], 5))
    k_j.append(prib_z[k].index(i) + 1)
    t_kob.append(t_sob[rfind(j_l, i)])
    if(t_kob[i-1] == t_nob[i-1]):
        t_kob[i-1] = -1
        t_obsl.append(-1)
    else:
        t_obsl.append(round(t_kob[i-1] - t_nob[i-1], 5))
    break

if(C_l[j_l.index(i)] > n):
    q_j.append(C_l[j_l.index(i)] - n)
else:
    q_j.append(0)

#print(j)
#print(t_z)
#print(q_j)
#print(t_och)
#print(t_nob)
#print(t_obsl)
#print(t_kob)
#print(k_j)

print("j;t_z;q_j;t_och;t_nob;t_obsl;t_kob;k_j")
for i in range(len(j)):
    print(f"{j[i]};{t_z[i]};{q_j[i]};{t_och[i]};{t_nob[i]};{t_obsl[i]};{t_kob[i]};{k_j[i]}")

k = []
N_k = []
t_zan = [0 for x in range(n)]
t_pr = []
delta_pr = []

for i in range(n):
    k.append(i+1)
    tmp = set()
    for t in range(100):
        if(prib_z[t][i] != 0):
            tmp.add(prib_z[t][i])
    N_k.append(len(tmp))

```

```

tmp = set()
for t in range(1, 100):
    if(prib_z[t][i] not in tmp):
        t_zan[i] += round(prib_time[t][i], 5)
        tmp.add(prib_z[t][i])
t_zan[i] = round(t_zan[i], 5)
if(t_zan[i] > max(t_sob)):
    t_zan[i] = round(max(t_sob), 5)
t_pr.append(round(max(t_sob) - t_zan[i], 5))
delta_pr.append(round(t_pr[i] / max(t_sob), 5))
#print(k)
#print(N_k)
#print(t_zan)
#print(t_pr)
#print(delta_pr)

print("k;N_k;t_zan;t_pr;delta_pr")
for i in range(len(k)):
    print(f"{k[i]};{N_k[i]};{t_zan[i]};{t_pr[i]};{delta_pr[i]}")

print(sum(delta_pr) / n)

J = max(j_l)
JF = Type.count(2)
JL = Type.count(3)

z = round(sum(C_l) / 100, 5)
_t_och = round(sum(t_och) / JF, 5)
_t_smo = 0
for i in range(JF):
    _t_smo += round(t_kob[i] - t_z[i], 5)

_t_smo = _t_smo / 100

print("J:", J)
print("JF: ", JF)
print("JL: ", JL)
print("z: ", z)
print("_t_och: ", _t_och)
print("_t_smo: ", _t_smo)

```