**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №1

з дисципліни «Моделювання та Аналіз Програмного Забезпечення» на тему:

«Методи створення моделей з використанням системи імітаційного моделювання SIMC»

**Виконав:**

Студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

Викладач Ж. К. Камінська

2024

МЕТОДИ СТВОРЕННЯ МОДЕЛЕЙ З ВИКОРИСТАННЯМ СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ SIMC

Мета роботи

Метою роботи є ознайомлення з принципами побудови імітаційних моделей систем масового обслуговування (СМО) з використанням системи імітаційного моделювання SIMC, вивчення системних констант, типів та змінних, групи процедур утворення системного та модельного середовища та динамічних моделей SIMC.

Опис типу даного Транзакт

Транзакт – це динамічний об’єкт бібліотеки SIMC, що позначає деяку подію у моделі та може бути визначений за допомогою багатьох параметрів.

Результати виконання роботи

Завдання 1

#include "../simc/simc.h"

int main()

{

  long long N = 19,

            D = 38,

            T = 190;

  pfacility factory;

  initlist(T);

  starttrace();

  initcreate(1, 0);

  newfac(factory, "\"Chocolate Factory\"");

  while (systime < T)

  {

    plan();

    switch (sysevent)

    {

    case 1:

      create(N);

      break;

    case 2:

      seize(factory);

      break;

    case 3:

      delayt(D);

      break;

    case 4:

      outfac(factory);

      break;

    case 5:

      destroy();

      break;

    }

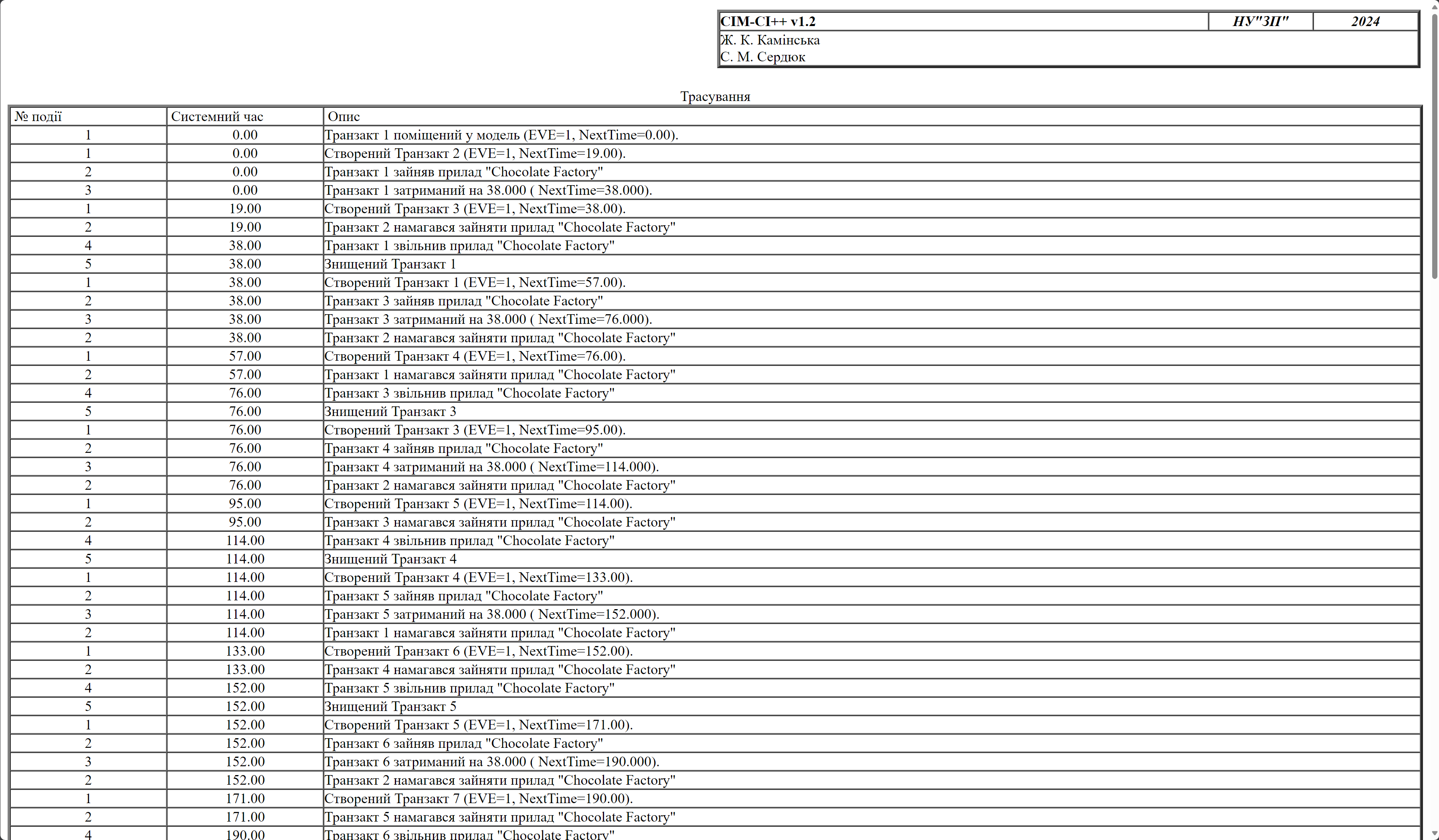
  }

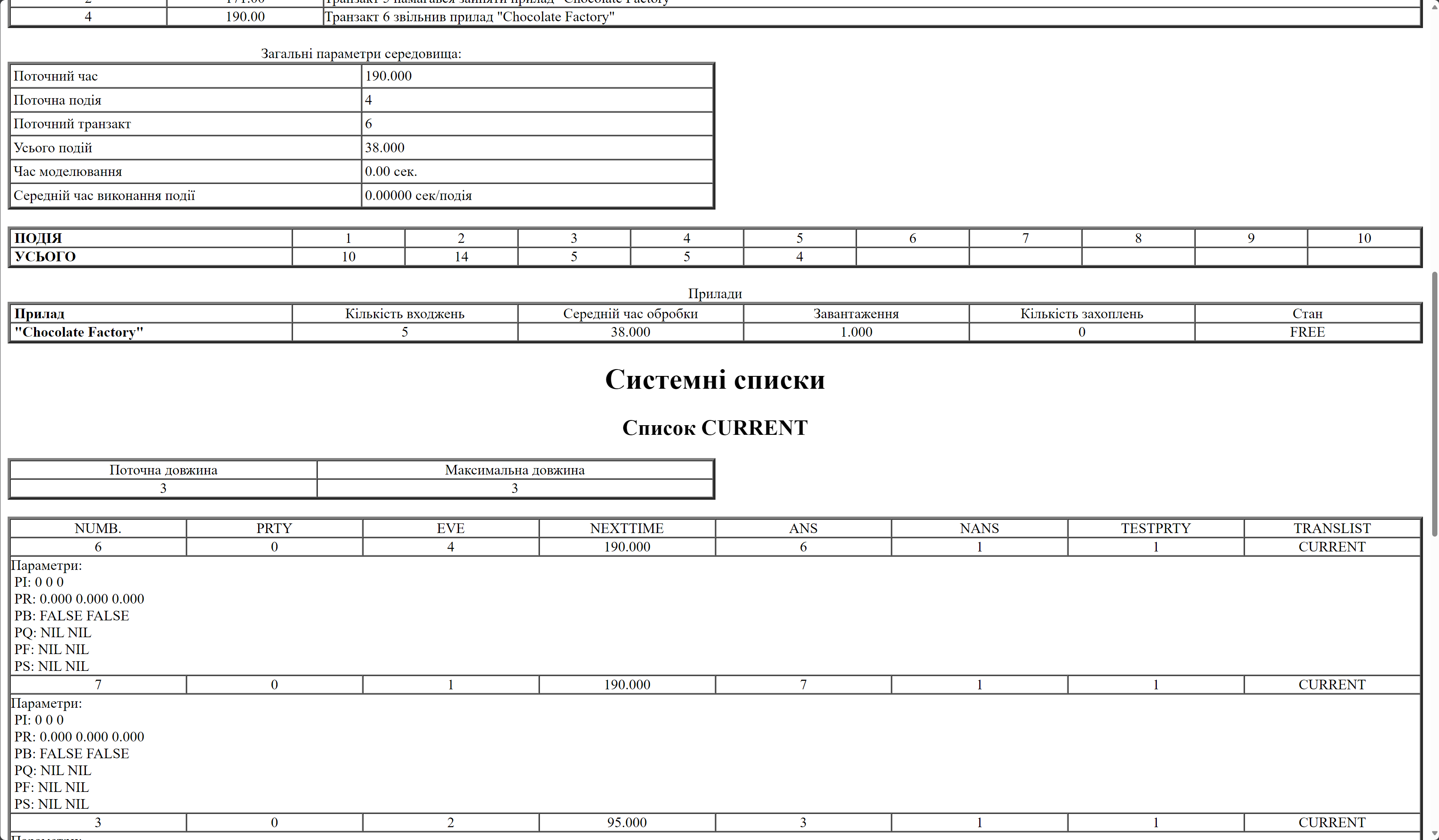
  printall();

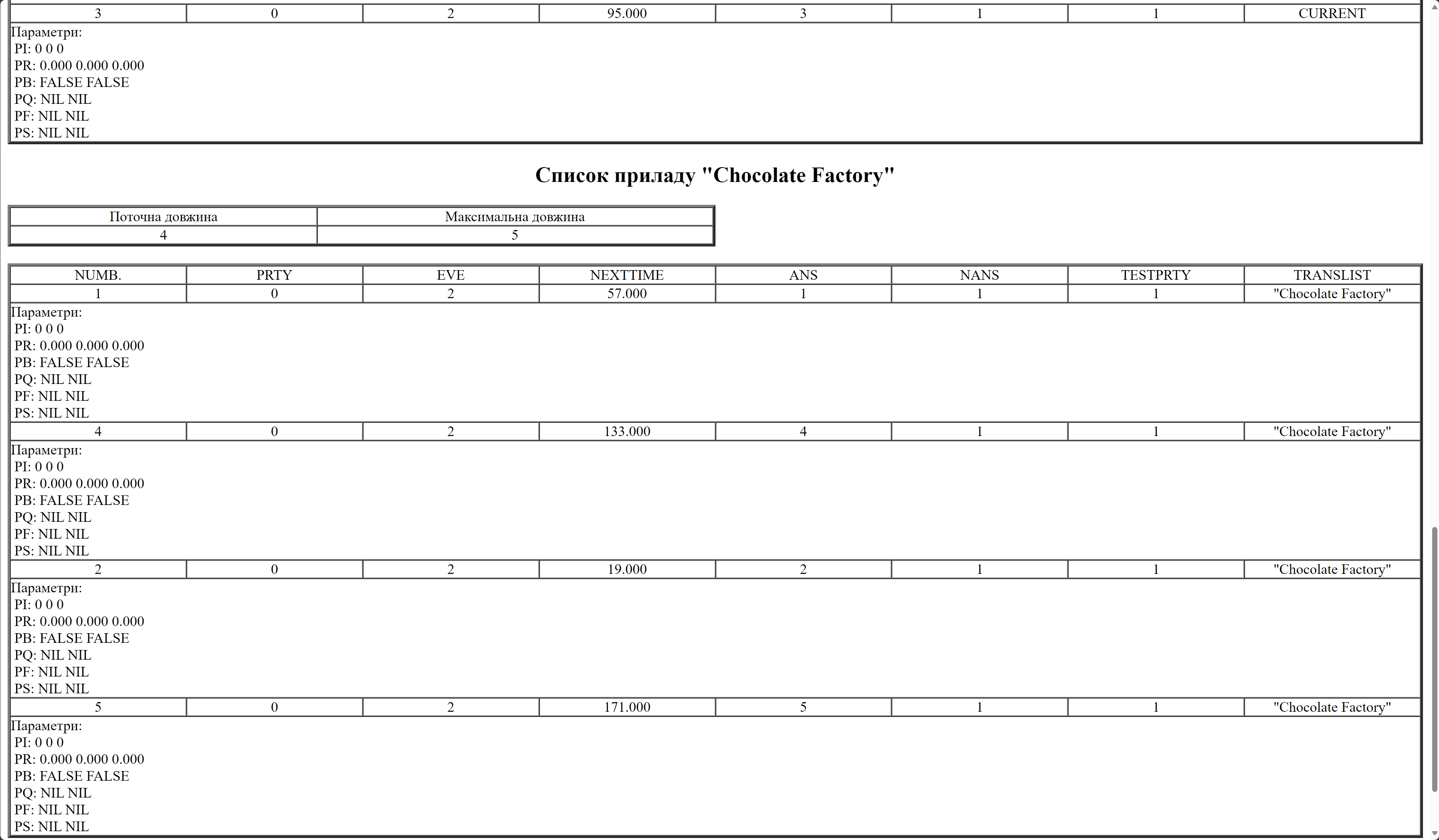
  return 0;

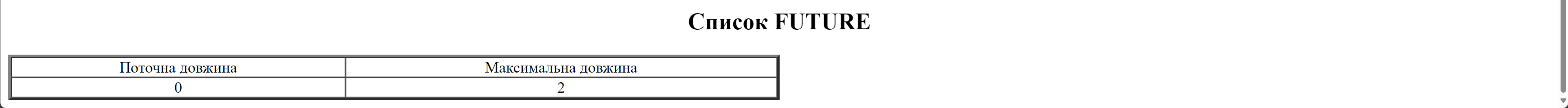
}

Нижче наведено результати моделювання у HTML файлі:









Завдання 2

#include "../simc/simc.h"

int main()

{

  long long B\_N = 19,

            B\_D = 38,

            W\_N = 29,

            W\_D = 57,

            T = 190;

  pfacility factory;

  initlist(T);

  starttrace();

  initcreate(1, 0);

  initcreate(6, 2);

  newfac(factory, "\"Chocolate Factory\"");

  while (systime < T)

  {

    plan();

    switch (sysevent)

    {

    case 1:

      create(B\_N);

      break;

    case 2:

      seize(factory);

      break;

    case 3:

      delayt(B\_D);

      break;

    case 4:

      outfac(factory);

      break;

    case 5:

      destroy();

      break;

    case 6:

      create(W\_N);

      break;

    case 7:

      seize(factory);

      break;

    case 8:

      delayt(W\_D);

      break;

    case 9:

      outfac(factory);

      break;

    case 10:

      destroy();

      break;

    }

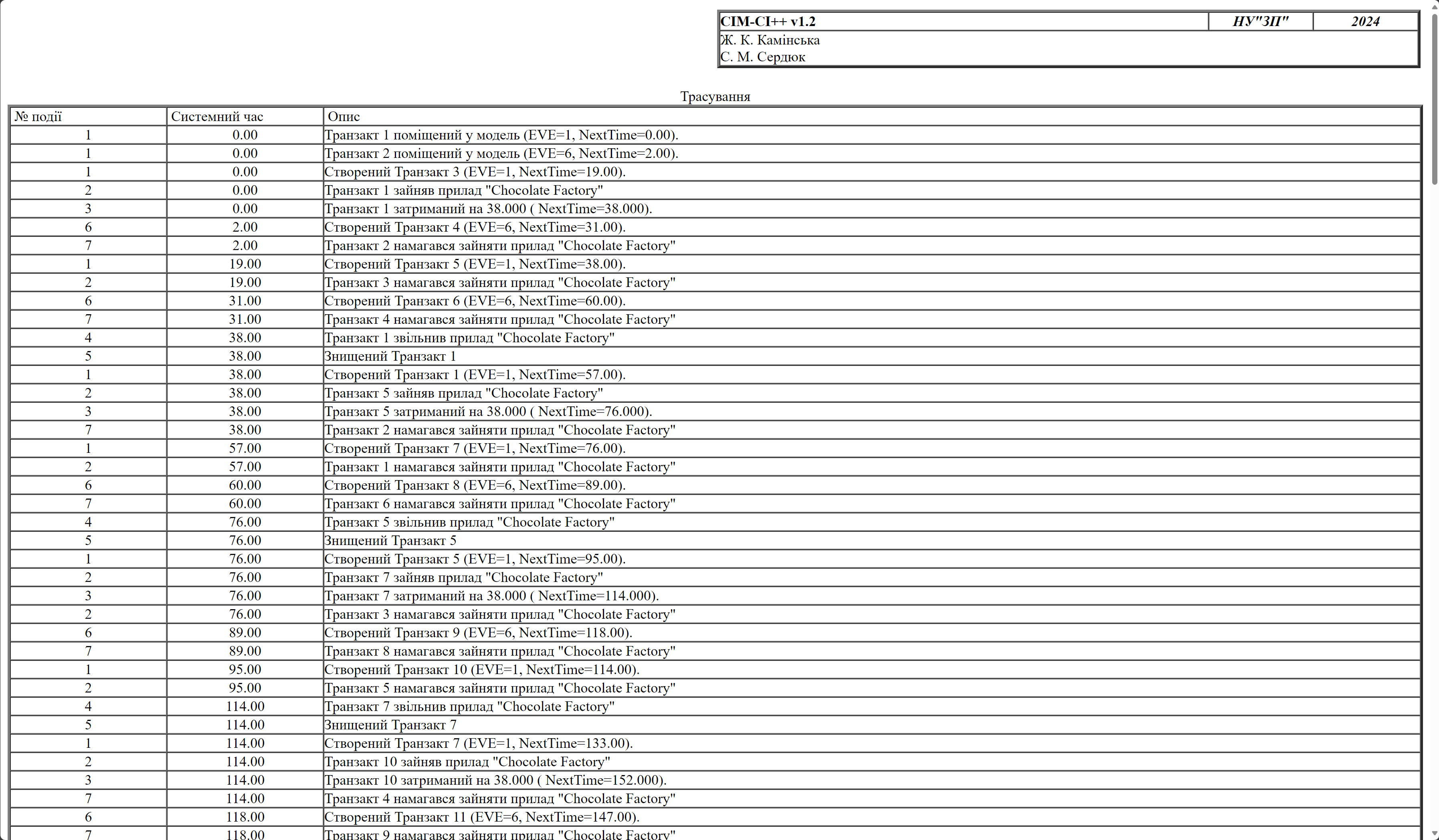
  }

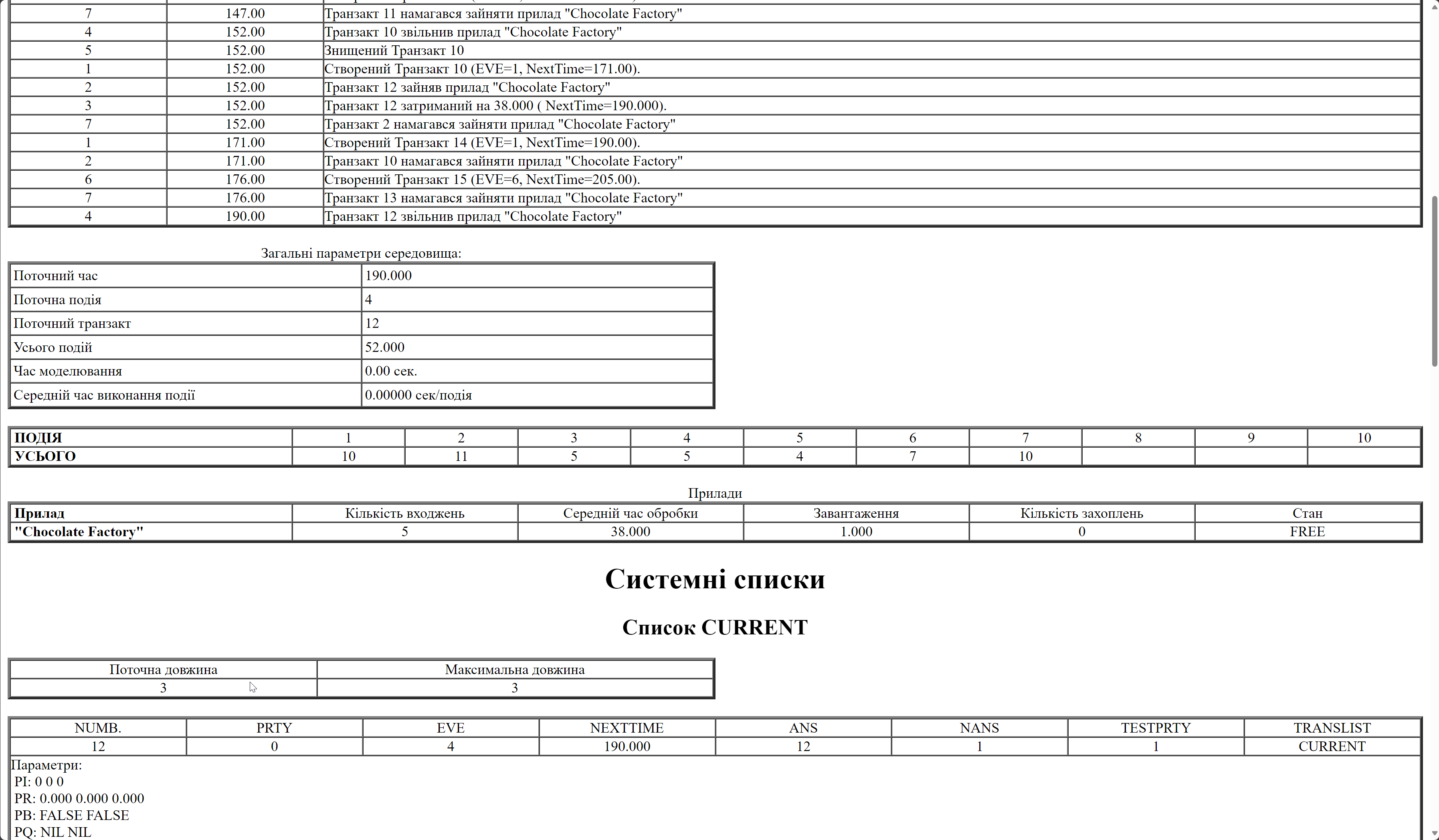
  printall();

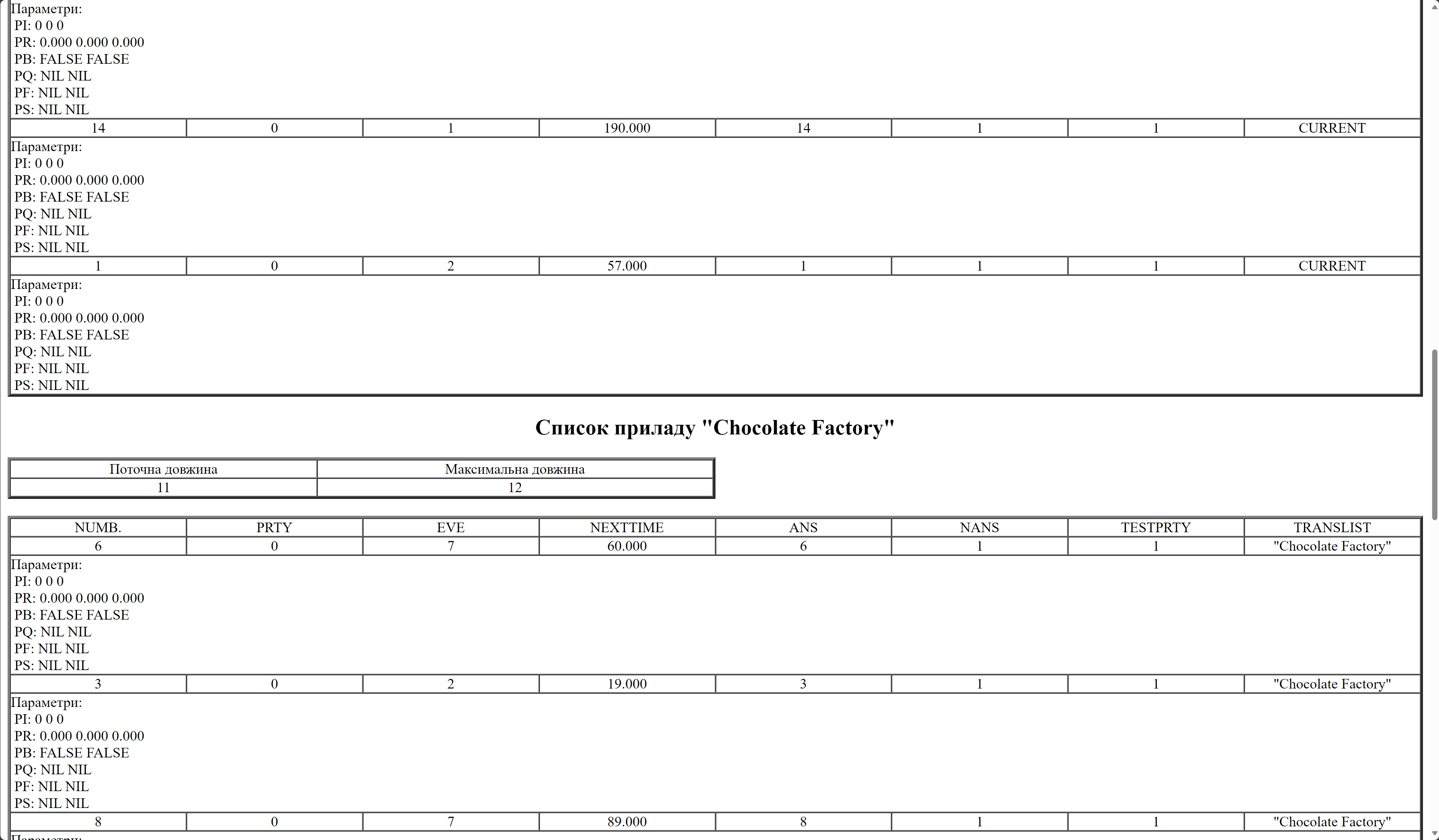
  return 0;

}

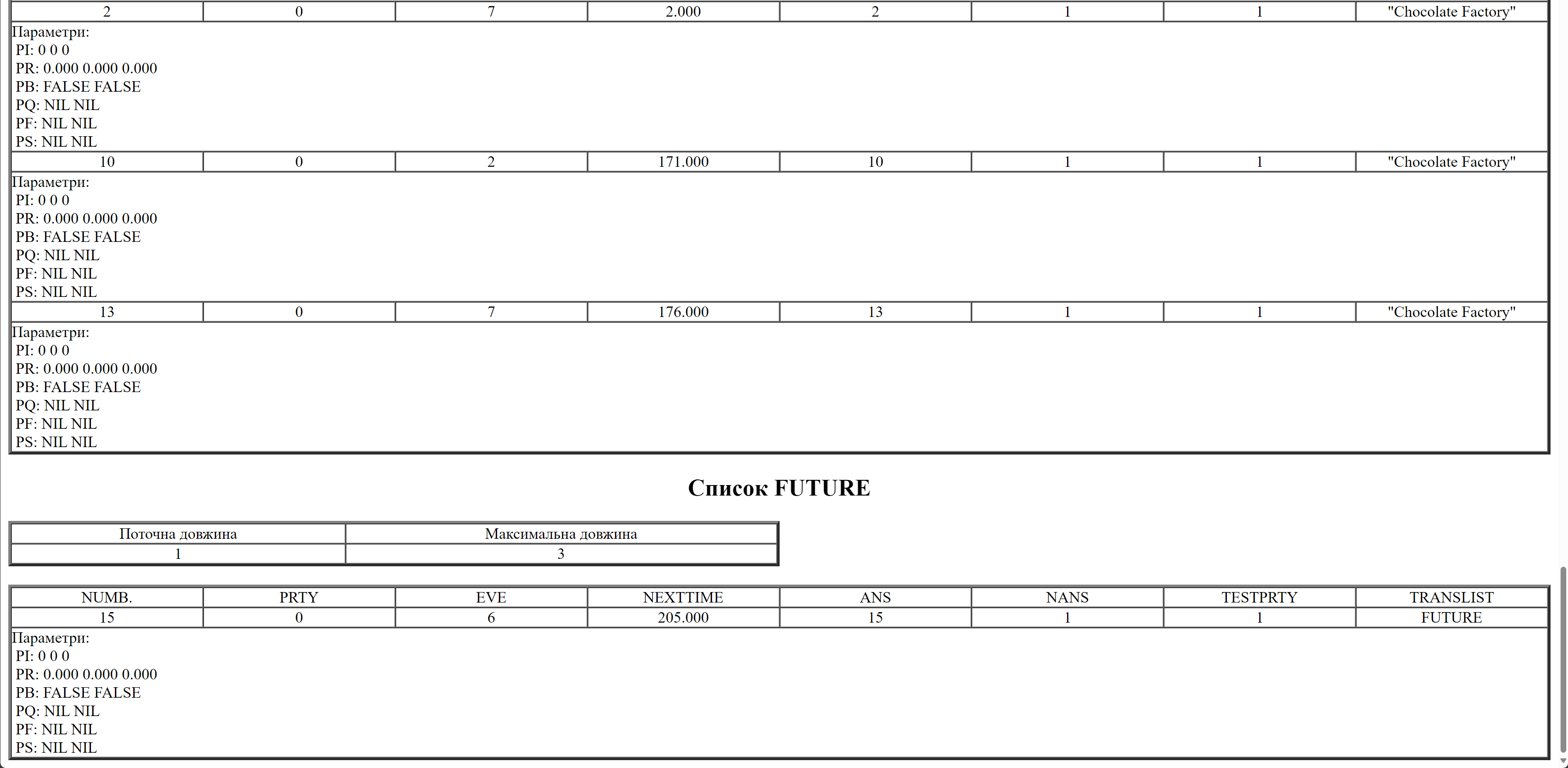
Нижче наведено результати моделювання у HTML файлі:











Індивідуальне завдання – Написання трасування

#include "../simc/simc.h"

int main()

{

  long long B\_N = 3,

            B\_D = 4,

            W\_N = 4,

            W\_D = 3,

            T = 8;

  pfacility factory;

  initlist(T);

  starttrace();

  initcreate(1, 0);

  initcreate(6, 0);

  newfac(factory, "\"Chocolate Factory\"");

  while (systime < T)

  {

    plan();

    switch (sysevent)

    {

    case 1:

      create(B\_N);

      break;

    case 2:

      seize(factory);

      break;

    case 3:

      delayt(B\_D);

      break;

    case 4:

      outfac(factory);

      break;

    case 5:

      destroy();

      break;

    case 6:

      create(W\_N);

      break;

    case 7:

      seize(factory);

      break;

    case 8:

      delayt(W\_D);

      break;

    case 9:

      outfac(factory);

      break;

    case 10:

      destroy();

      break;

    }

  }

  printall();

  return 0;

}

Нижче наведено таблицю трасування:

|  |  |  |  |
| --- | --- | --- | --- |
| **Case #** | **Case Description** | **System Time** | **List Name** |
| 1 | Транзакт 1. З'являється в моделі. NT=0 | 0 | Delist |
| 6 | Транзакт 2. З'являється в моделі. NT=0 | 0 | Delist |
| 1 | Транзакт 3. Транзакт створено. NT=3 | 0 | Delist |
| 6 | Транзакт 4. Транзакт створено. NT=4 | 0 | Delist |
| 2 | Транзакт 1. Зайняв прилад | 0 | Current |
| 3 | Транзакт 1. Транзакт затримано. NT=4 | 0 | Future |
| 7 | Транзакт 2. Намагався зайняти прилад | 0 | Facility List |
| 1 | Транзакт 5. Транзакт створено. NT=6 | 3 | Delist |
| 2 | Транзакт 3. Намагався зайняти прилад | 3 | Facility List |
| 4 | Транзакт 1. Звільнив прилад | 4 | Current |
| 5 | Транзакт 1. Транзакт знищено | 4 | Delist |
| 6 | Транзакт 6. Транзакт створено. NT=8 | 4 | Delist |
| 7 | Транзакт 4. Зайняв прилад | 4 | Current |
| 8 | Транзакт 4. Транзакт затримано. NT=7 | 4 | Future |
| 7 | Транзакт 2. Намагався зайняти прилад | 4 | Facility List |
| 1 | Транзакт 7. Транзакт створено. NT=9 | 6 | Delist |
| 2 | Транзакт 5. Намагався зайняти прилад | 6 | Facility List |
| 9 | Транзакт 4. Звільнив прилад | 7 | Current |
| 10 | Транзакт 4. Транзакт знищено | 7 | Delist |
| 2 | Транзакт 2. Зайняв прилад | 7 | Current |
| 3 | Транзакт 2. Транзакт затримано. NT=11 | 7 | Future |
| 6 | Транзакт 8. Транзакт створено. NT=12 | 8 | Delist |