Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №8

Тема работы: Сортировка двумерных массивов

Выполнил

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc89619629)

[2 Методика решения 4](#_Toc89619630)

[2.1 Заполнение матрицы 4](#_Toc89619631)

[2.2 Использование циклов и сортировки 4](#_Toc89619632)

[3 Текстовый алгоритм решения задачи 5](#_Toc89619633)

[4 Структура данных 7](#_Toc89619634)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc89619635)

[6 Схема алгоритма решения задачи по методу Дамке 11](#_Toc89619636)

[7 Схема алгоритма решения задачи по методу Насси-Шнейдермана 14](#_Toc89619637)

[8 Результаты расчетов 16](#_Toc89619638)

[Приложение А. Исходный код 17](#_Toc89619639)

[Приложение Б. Тестовые наборы 20](#_Toc89619640)

# Постановка задачи

Дана матрица X[9,10]. Упорядочить элементы столбцов матрицы по возрастанию, а сами столбцы - по возрастанию произведения чётных элементов.

Вывести в виде:

1. Изначальная матрица.
2. Произведения чётных элементов.
3. Отсортированная матрица.

# Методика решения

## Заполнение матрицы

Для заполнения изначальной матрицы с 10 строками и 10 столбцами можно применить генератор равномерно-распределённых чисел. При этом заполнить необходимо лишь 9 строк, так как десятая зарезервирована для хранения данных из последующих вычислений.

## Использование циклов и сортировки

Чтобы вычислить произведения чётных элементов в столбцах матрицы, нужно использовать два последовательных цикла. Первый цикл должен перебирать столбцы, а второй, с предусловием, накапливать в десятом элементе каждого столбца произведение его чётных элементов (каждого второго элемента, начиная со второго).

Чтобы отсортировать значения в столбцах по возрастанию, необходимо применить три вложенных цикла: один должен перебирать столбцы, а второй и третий – осуществлять сортировку пузырьком. При этом нельзя включать в сортировку десятый элемент каждого столбца, поскольку он в изначальной матрице не существует и там хранятся произведения чётных элементов.

Сортировка столбцов по возрастанию сохранённых произведений происходит в три вложенных цикла. Два цикла используются для сортировки сохранённых произведений пузырьком, а третий – для того, чтобы обменять местами не только произведения, но и столбцы, в которых эти произведения являются десятыми элементами.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод А[1..10, 1..10] |
|  | J:=1 |
|  | Начало цикла А1. Проверка выполнения условия (J<=10). Если условие истинно, перейти к шагу 4, иначе – к шагу 12 |
|  | A[10, J]:= 1 |
|  | I:=2 |
|  | Начало цикла А2. Идти к шагу 7. |
|  | A[10, J]:= A[10, J] \* A[I, J] |
|  | I:=I+2 |
|  | Конец цикла А2. Проверка выполнения условия (I = 10). Если условие истинно, перейти к шагу 10, иначе – к шагу 6 |
|  | J:=J+1 |
|  | Конец цикла А1. Идти к шагу 3 |
|  | J:=1 |
|  | Начало цикла А3. Проверка выполнения условия (J<=10). Если условие истинно, перейти к шагу 14, иначе – к шагу 26 |
|  | I:=1 |
|  | Начало цикла А4. Проверка выполнения условия (I<=9). Если условие истинно, перейти к шагу 16, иначе – к шагу 24 |
|  | K:=1 |
|  | Начало цикла А5. Проверка выполнения условия (K<=9-I). Если условие истинно, перейти к шагу 18, иначе – к шагу 22 |
|  | Проверка выполнения условия (A[K, J] > A[K + 1, J]). Если условие истинно, перейти к шагу 19, иначе – к шагу 20 |
|  | Swap(A[K, J], A[K+1, J]) |
|  | K:=K+1 |
|  | Конец цикла А5. Идти к шагу 17 |
|  | I:=I+1 |
|  | Конец цикла А4. Идти к шагу 15 |
|  | J:=J+1 |
|  | Конец цикла А3. Идти к шагу 13 |
|  | J:=1 |
|  | Начало цикла А6. Проверка выполнения условия (J<=10). Если условие истинно, перейти к шагу 28, иначе – к шагу 40 |
|  | K:=1 |
|  | Начало цикла А7. Проверка выполнения условия (K<=10-J). Если условие истинно, перейти к шагу 30, иначе – к шагу 38 |

Продолжение Таблицы 1

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Проверка выполнения условия (A[10, K] > A[10, K + 1]). Если условие истинно, перейти к шагу 31, иначе – к шагу 36 |
|  | I:=1 |
|  | Начало цикла А8. Проверка выполнения условия (I<=10). Если условие истинно, перейти к шагу 33, иначе – к шагу 36 |
|  | Swap(A[I, K], A[I, K + 1]) |
|  | I:=I+1 |
|  | Конец цикла А8. Идти к шагу 32 |
|  | K:=K+1 |
|  | Конец цикла А7. Идти к шагу 29 |
|  | J:=J+1 |
|  | Конец цикла А6. Идти к шагу 27 |
|  | Вывод А[1..10, 1..10] |
|  | Останов. |

# Структура данных

Таблица 2 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| A | Array[1..10, 1.10] of String | Двумерный массив (матрица) |
| I | Integer | Номер строки |
| J | Integer | Номер столбца |
| K | Integer | Используемая в сортировке пузырьком переменная, которая может обозначать номер строки или номер столбца |
| Temp | Integer | Переменная, сохраняющая значение элемента при переприсваиваниях в сортировке |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

|  |
| --- |
|  |
| Рисунок 3 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

# Схема алгоритма решения задачи по методу Дамке

|  |
| --- |
|  |
|  |
| Рисунок 4 – Схема алгоритма решения задачи по методу Дамке |

|  |
| --- |
|  |
|  |
| Рисунок 5 – Схема алгоритма решения задачи по методу Дамке |

|  |
| --- |
|  |
|  |
| Рисунок 6 – Схема алгоритма решения задачи по методу Дамке |

# Схема алгоритма решения задачи по методу Насси-Шнейдермана

|  |
| --- |
|  |
|  |
| Рисунок 7 – Схема алгоритма решения задачи по методу Насси-Шнейдермана |

|  |
| --- |
|  |
|  |
| Рисунок 8 – Схема алгоритма решения задачи по методу Насси-Шнейдермана |

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

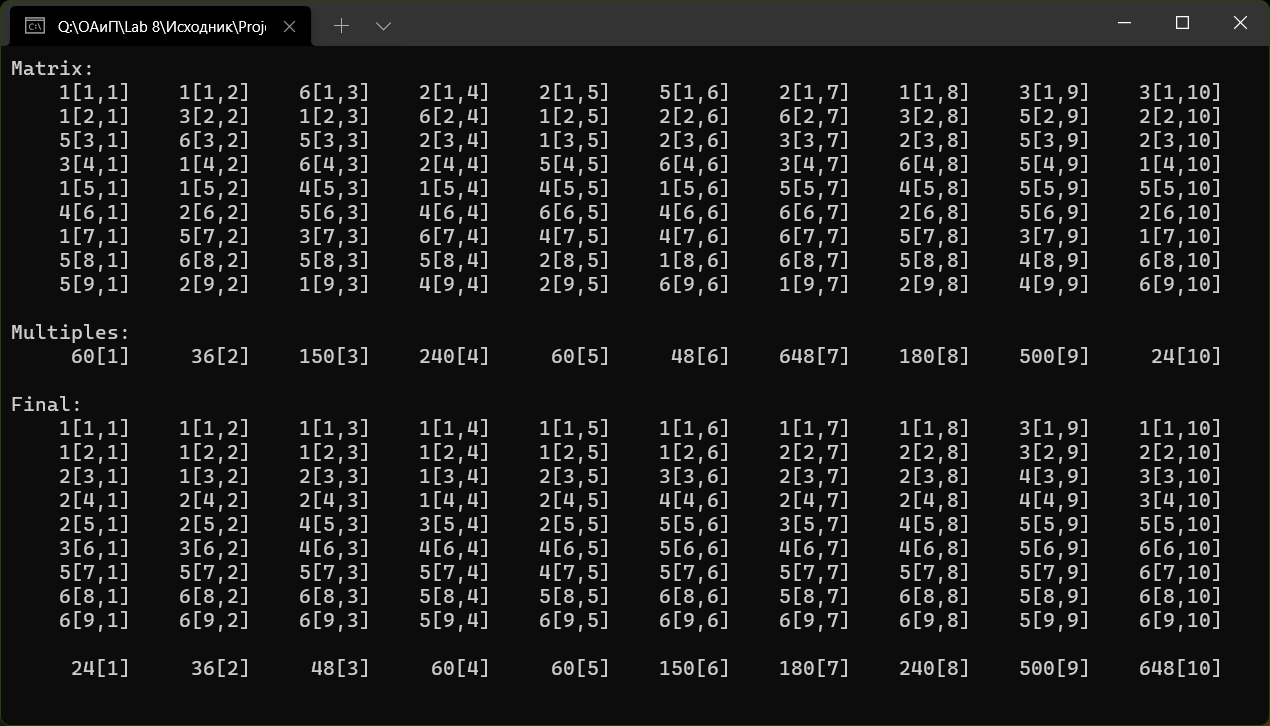


Рисунок 9 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы (постановка 1)

Program Lab8;

{This program sorts the matrix with some conditions}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare vars

Var

A: Array[1..10, 1..10] of Integer;

J, I, K, Temp: Integer;

//A - start matrix

//I - number of line

//J - number of column

//K - auxilary member during sort

//Temp - saving member during sort

Begin

Randomize;

//Generate the matrix and display it

Write('Matrix:');

For I:= 1 to 9 do

Begin

WriteLn;

For J:= 1 to 10 do

Begin

A[I, J]:= Random(6) + 1;

Write(A[I, J]:5, '[', I, ',', J, ']');

End;

End;

WriteLn;

WriteLn;

//Calculate multiples and display it

WriteLn('Multiples:');

For J:= 1 to 10 do

Begin

A[10, J]:= 1;

I:= 2; (\* First odd \*)

Repeat

A[10, J]:= A[10, J] \* A[I, J];

I:= I + 2; (\* Other odds \*)

Until I = 10;

Write(A[10, J]:7, '[', J, ']');

End;

WriteLn;

WriteLn;

//Sort elements in columnts excluding line of extended items

For J:= 1 to 10 Do

For I:= 1 to 9 Do

For K:= 1 to 9 - I Do

If (A[K, J] > A[K + 1, J]) then

Begin

Temp:= A[K, J];

A[K, J]:= A[K + 1, J];

A[K + 1, J]:= Temp;

End;

//Sort line of extended items

For J:= 1 to 10 do

For K:= 1 to 10 - J do

If (A[10, K] > A[10, K + 1]) then

//If extended item is moved, move the all column

For I:= 1 to 10 do

Begin

Temp:= A[I, K];

A[I, K]:= A[I, K + 1];

A[I, K + 1]:= Temp;

End;

//Display final sorted matrix

Write('Final:');

For I:= 1 to 9 do

Begin

WriteLn;

For J:= 1 to 10 do

Write(A[I, J]:5, '[', I, ',', J, ']');

End;

WriteLn;

WriteLn;

//Display new multiple places

For J:= 1 to 10 do

Write(A[10, J]:7, '[', J, ']');

ReadLn;

End.

Приложение Б

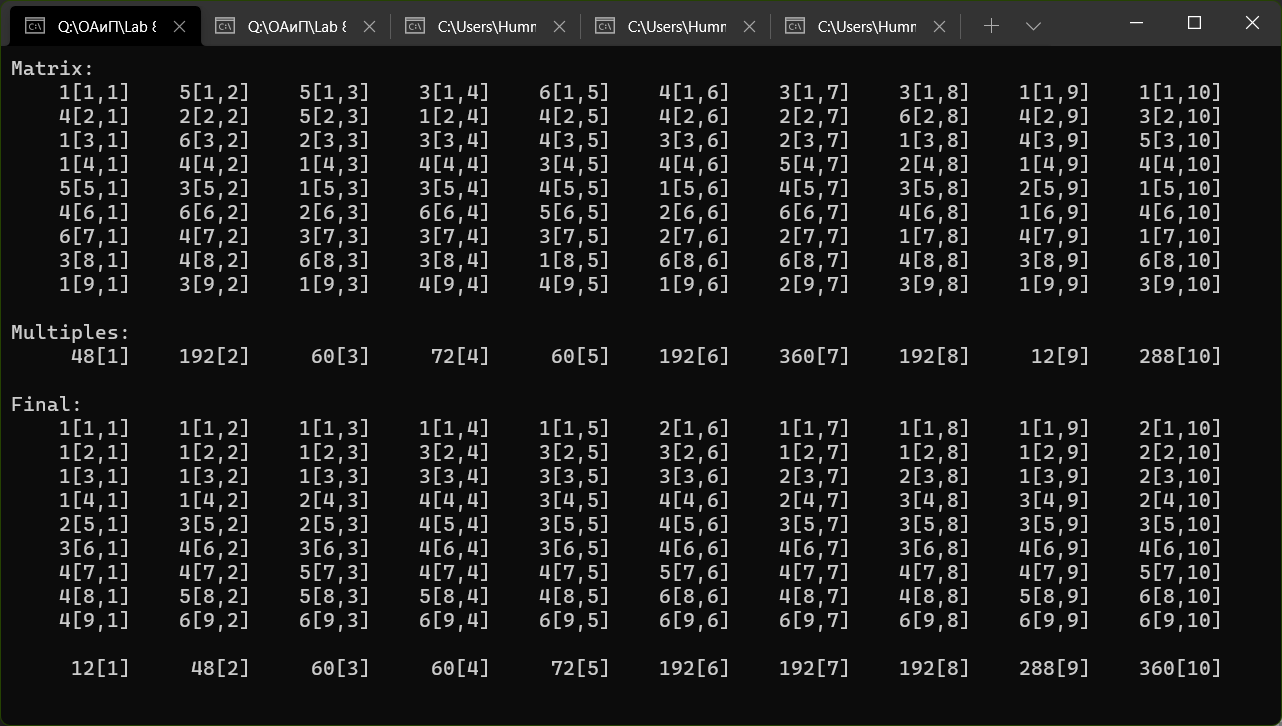
(обязательное)

Тестовые наборы

Тестовая ситуация 1: проверка результатов

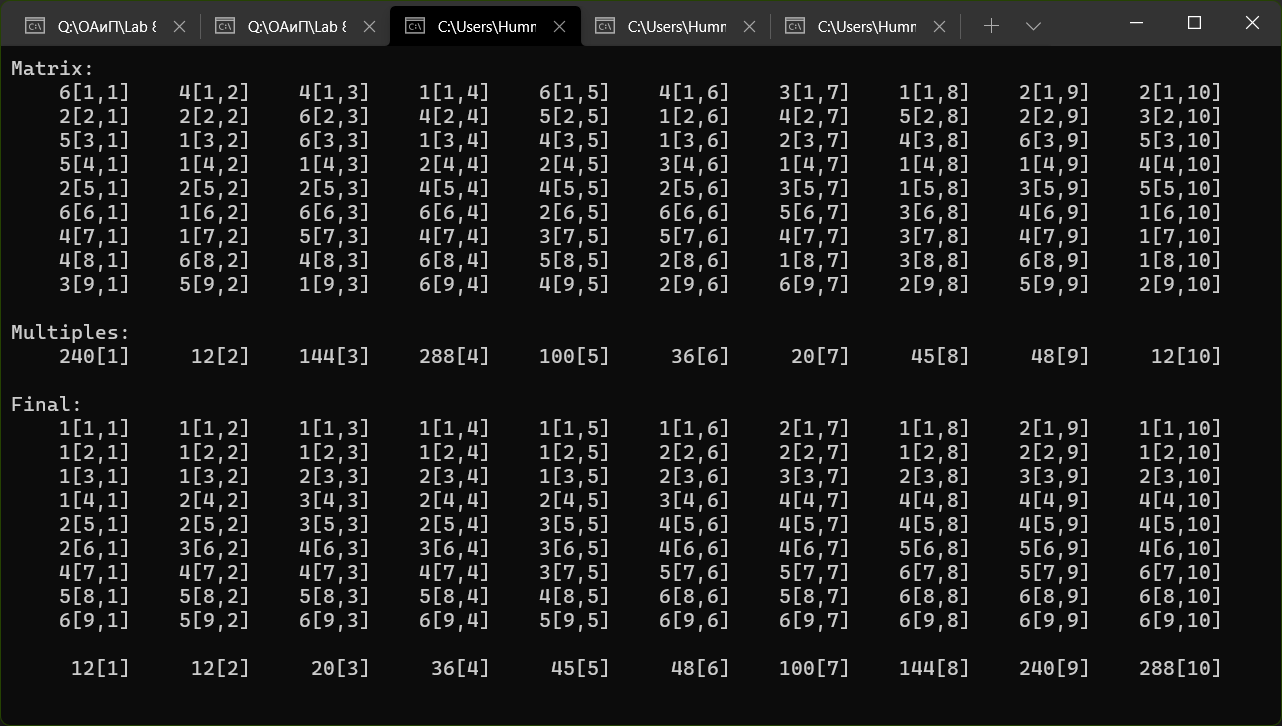
Тестовая ситуация: верно ли программа сортирует рандомно сгенерированную матрицу.

Матрица 1



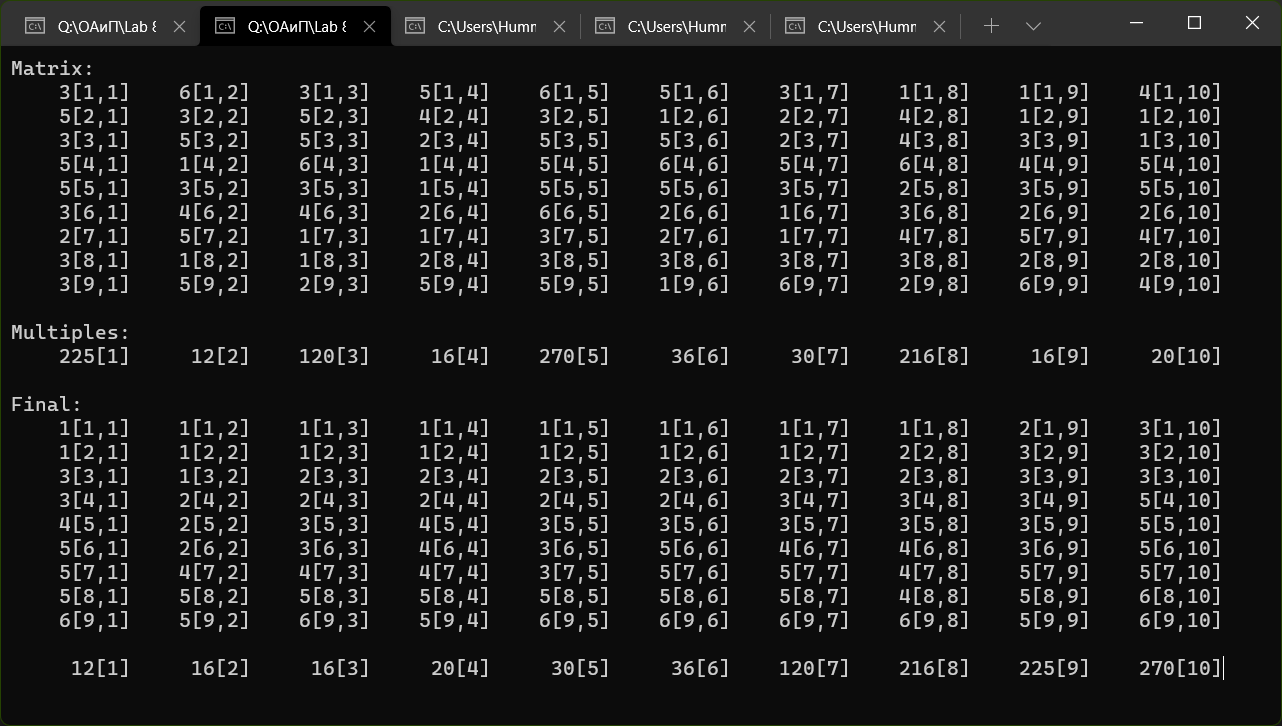
Очевидно, что все элементы столбцов отсортированы в порядке возрастания, а сами столбцы – в порядке возрастания произведений чётных элементов, что удовлетворяет условию задачи.

Матрица 2



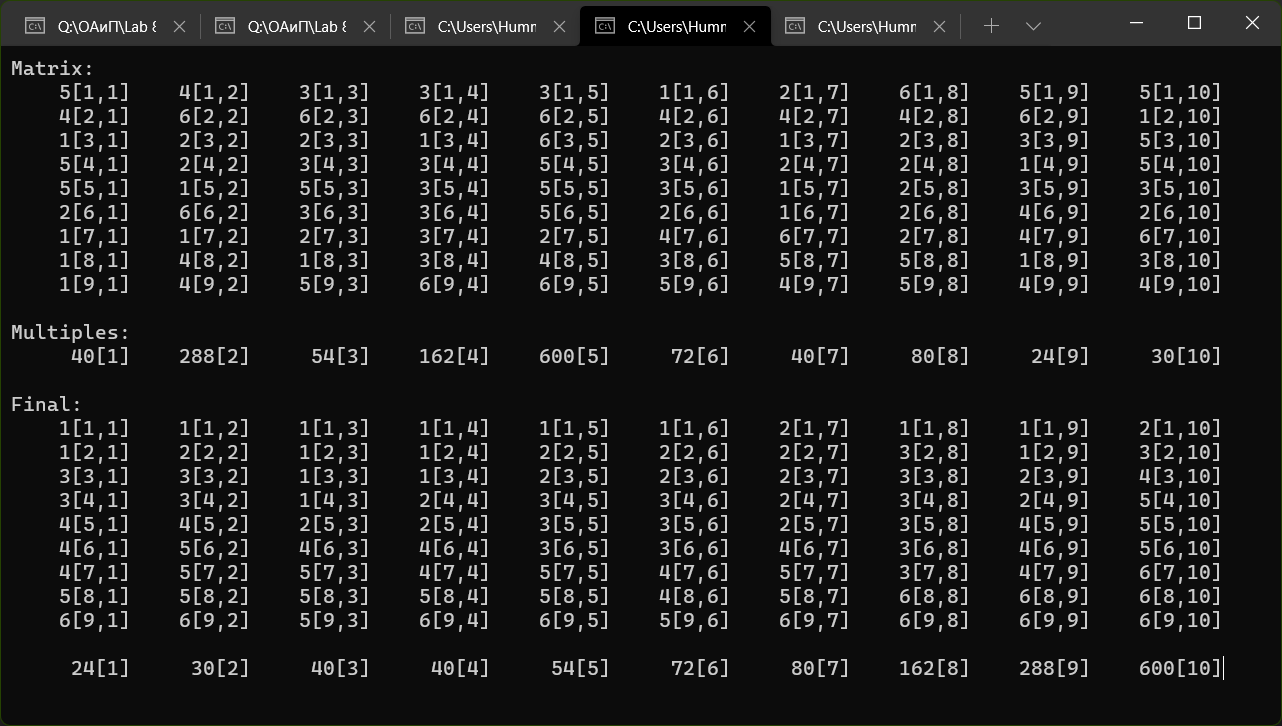
Очевидно, что все элементы столбцов отсортированы в порядке возрастания, а сами столбцы – в порядке возрастания произведений чётных элементов, что удовлетворяет условию задачи.

Матрица 3



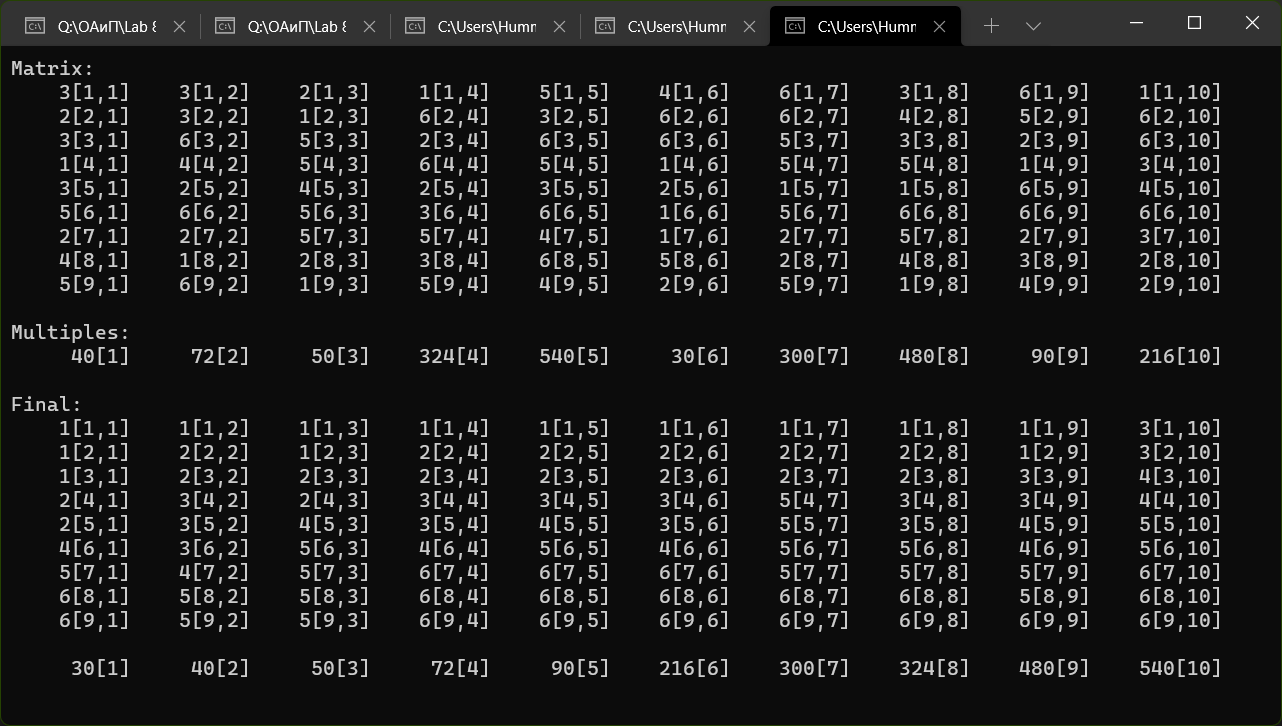
Очевидно, что все элементы столбцов отсортированы в порядке возрастания, а сами столбцы – в порядке возрастания произведений чётных элементов, что удовлетворяет условию задачи.

Матрица 4



Очевидно, что все элементы столбцов отсортированы в порядке возрастания, а сами столбцы – в порядке возрастания произведений чётных элементов, что удовлетворяет условию задачи.

Матрица 5



Очевидно, что все элементы столбцов отсортированы в порядке возрастания, а сами столбцы – в порядке возрастания произведений чётных элементов, что удовлетворяет условию задачи.