

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)**

Факультет Информатика и вычислительная техника

(наименование факультета)

Кафедра Кибербезопасность информационных систем

(наименование кафедры)

**ОТЧЕТ**

по лабораторным работам по предмету: Операционные системы

Выполнил:

ст. гр. ВКБ31 Котелевец К.А.

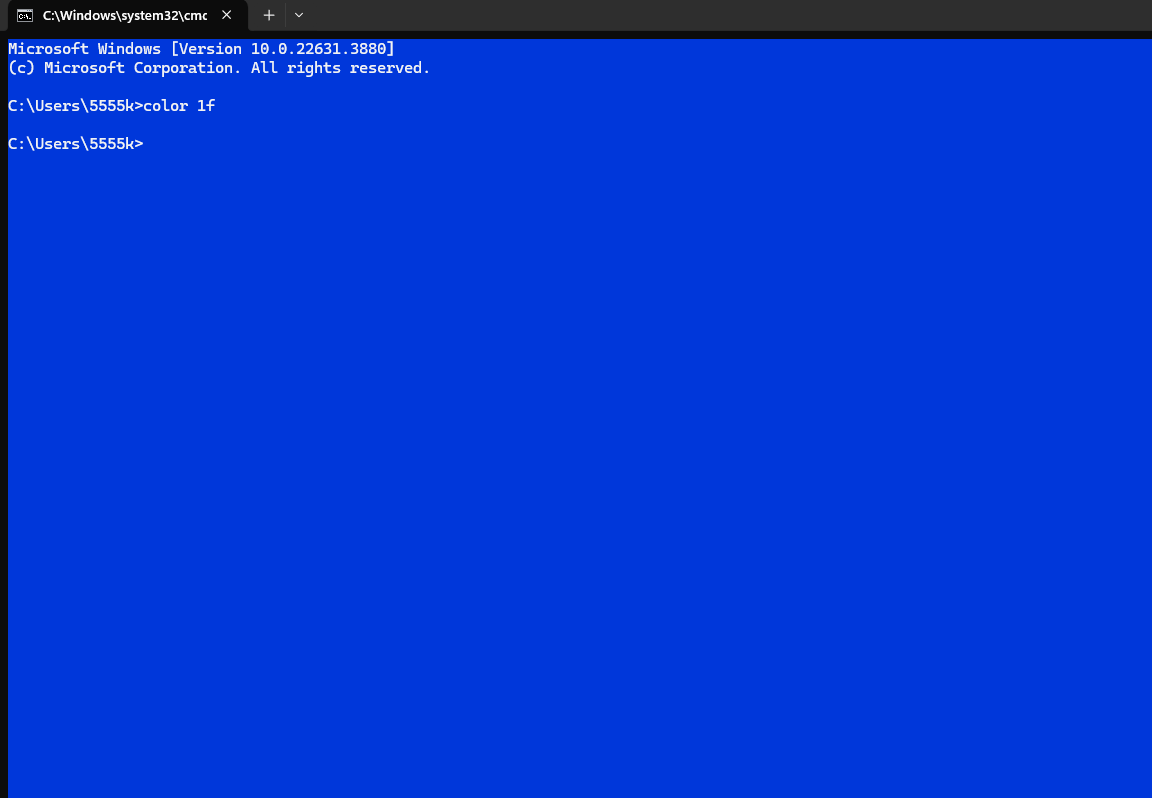
Ростов-на-Дону

2024

**ЛАБОРАТОРНАЯ РАБОТА №1 ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОС MS WINDOWS**

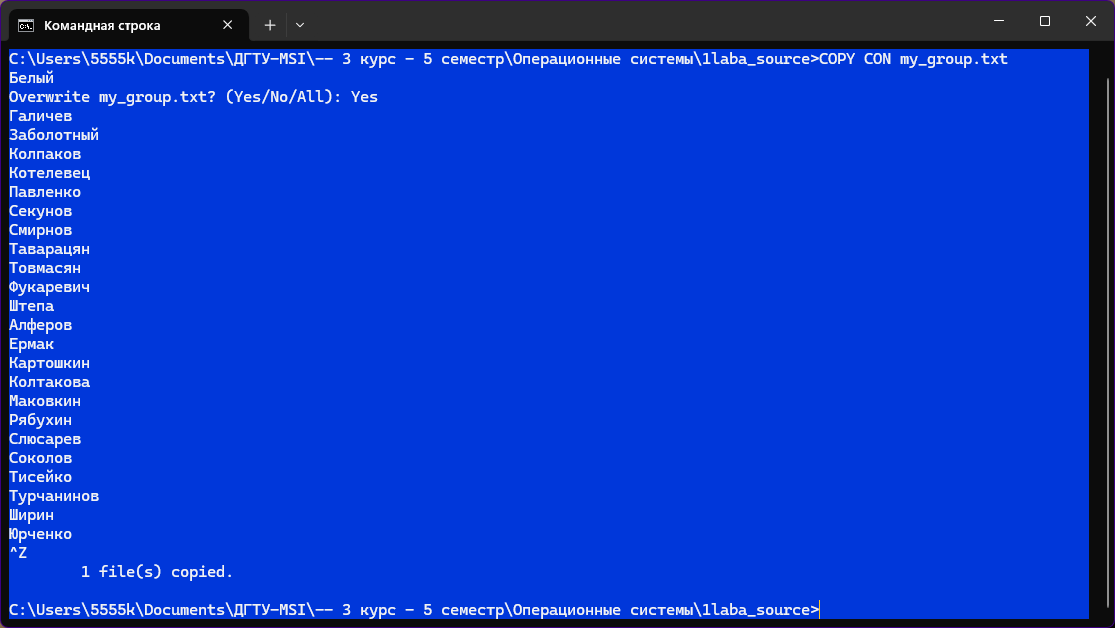
**Цель работы** – знакомство с возможностями интерпретатора командной строки и командами MS Windows

Задание 1. Увеличить размер окна интерпретатора и задать цвет фона и цвет шрифта (рекомендуется синий фон и белый шрифт)

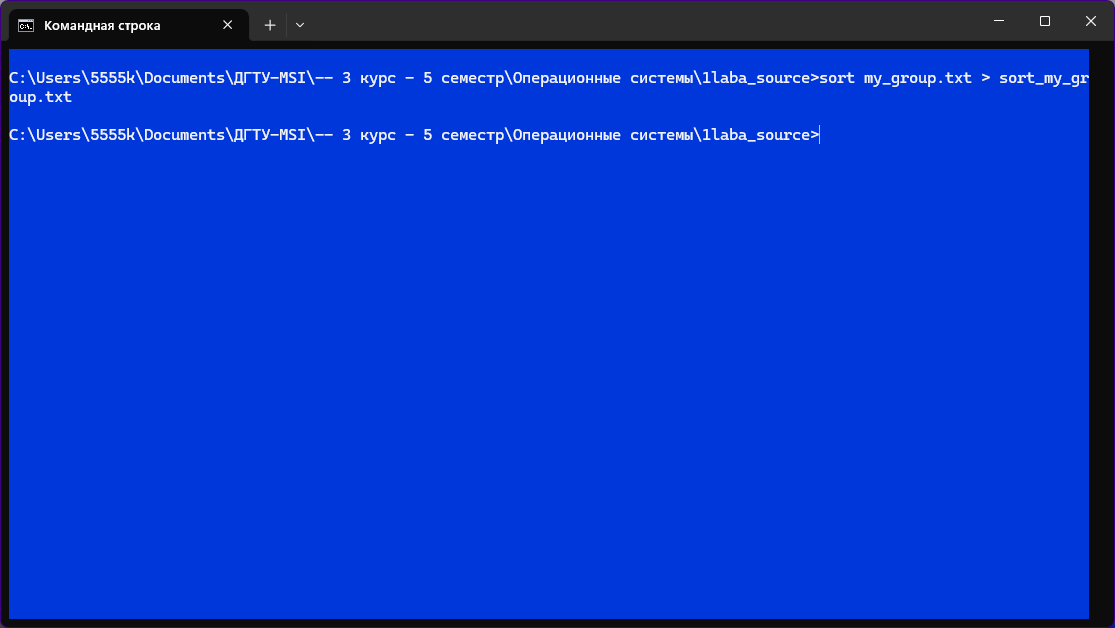


Задание 2. Создать список фамилий студентов группы. Отсортировать список в алфавитном порядке и сохранить его в новом файле.

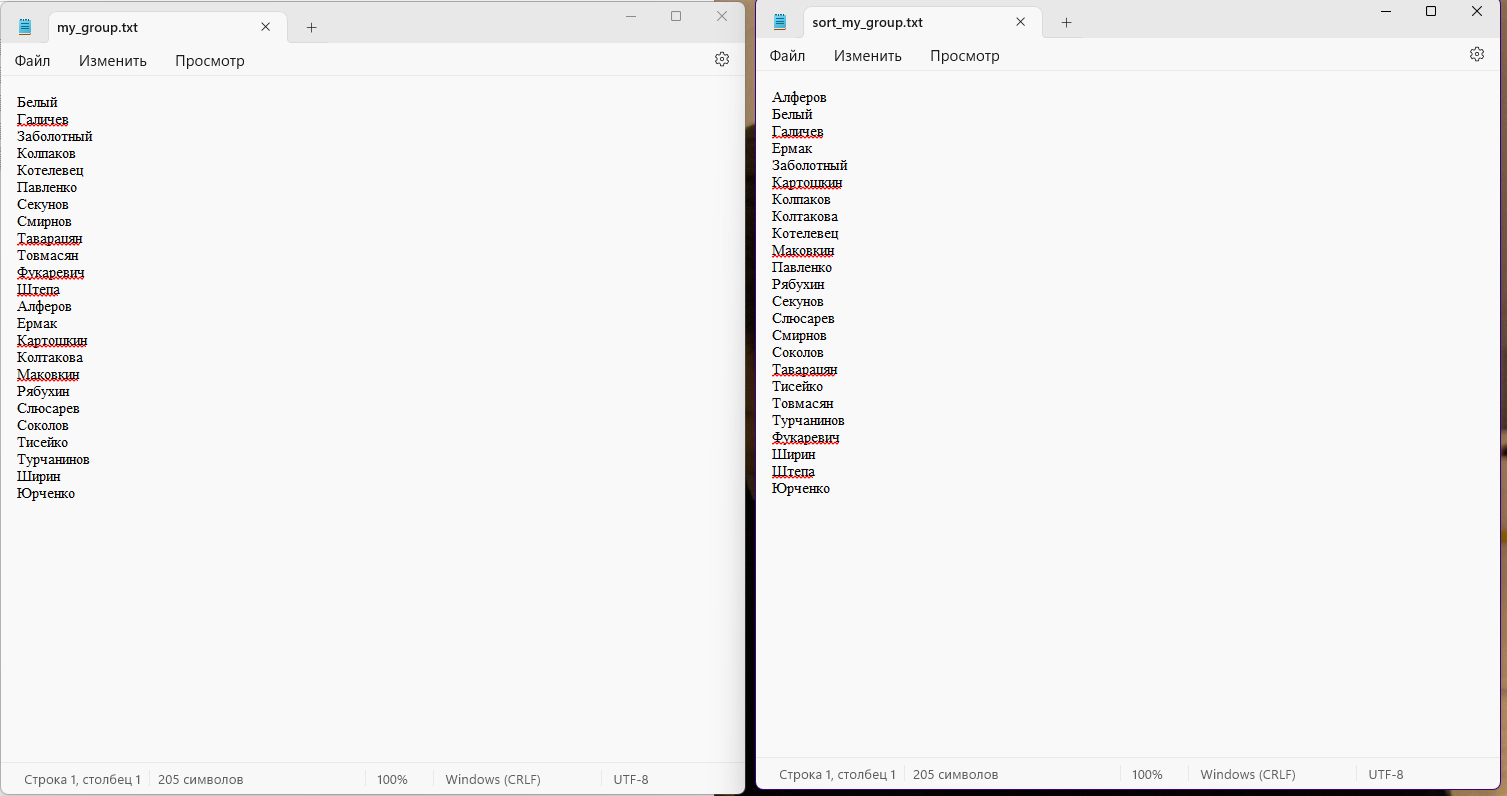
Создадим файл, в котором будут фамилий студентов моей группы, с помощью команды COPY CON my\_group.txt. Впишем все фамилии в данный файл в разном порядке.



Теперь отсортируем данный список в алфавитном порядке в новый текстовый файл sort\_my\_group.txt.

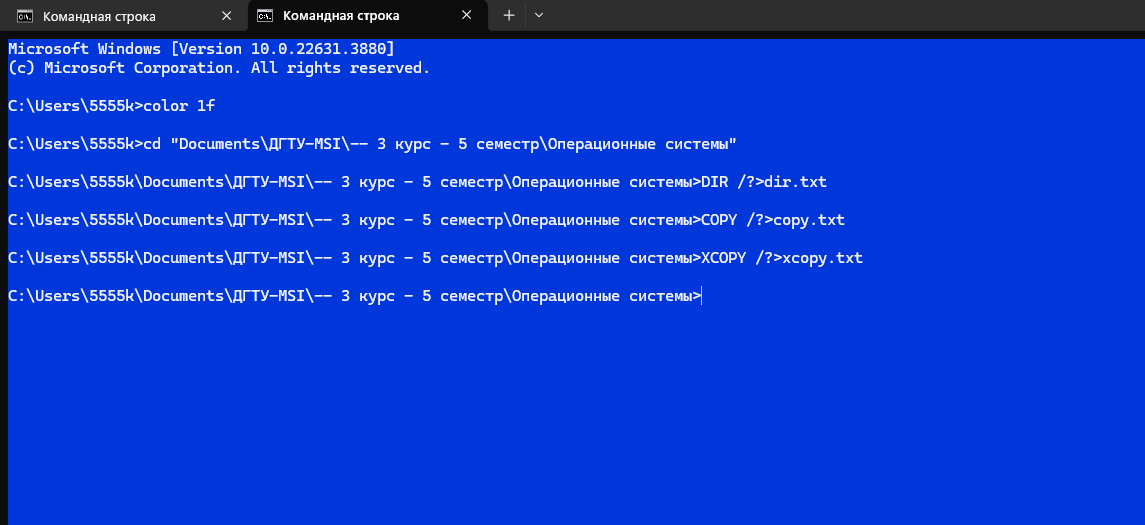


Результат:

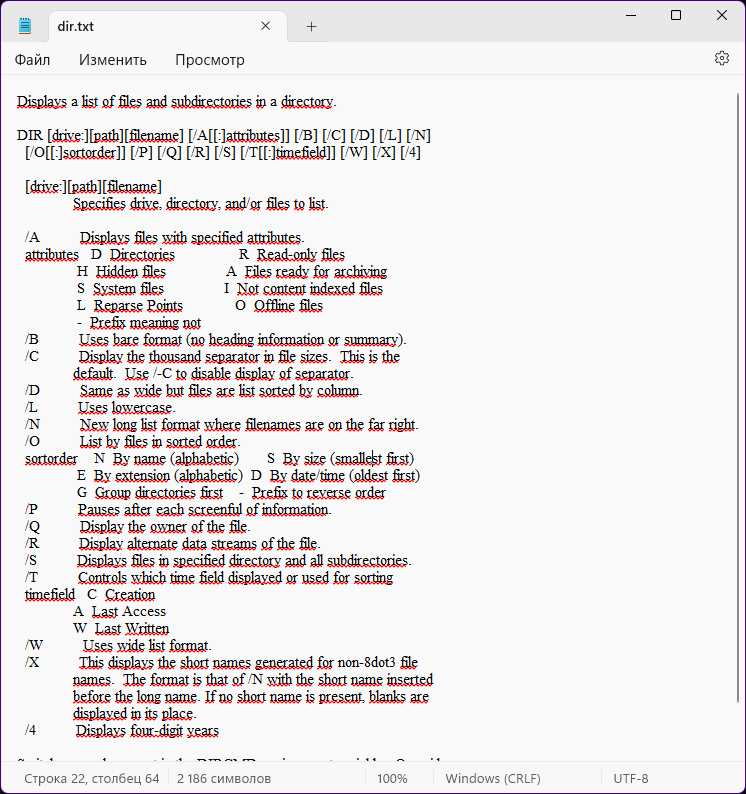


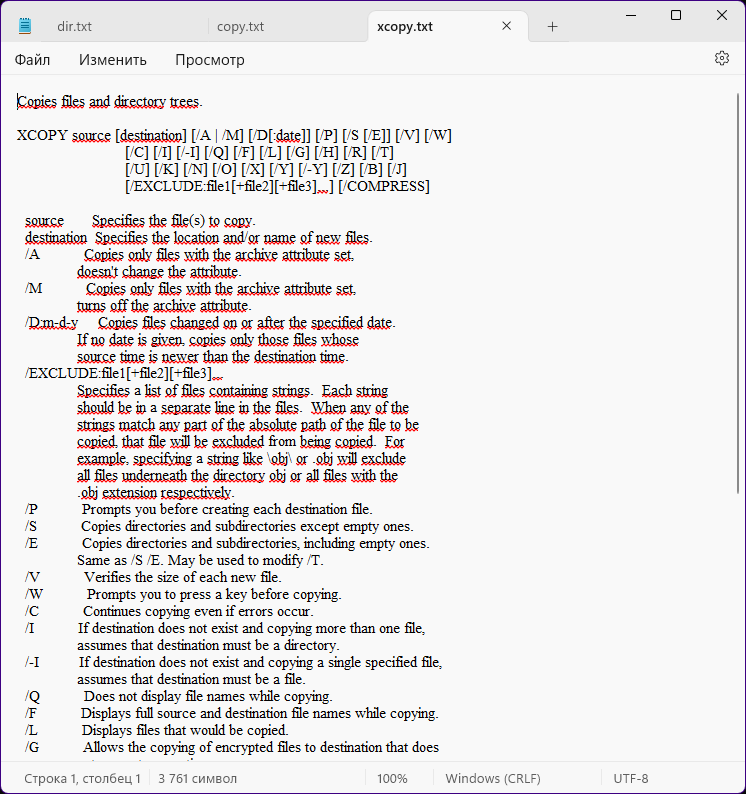
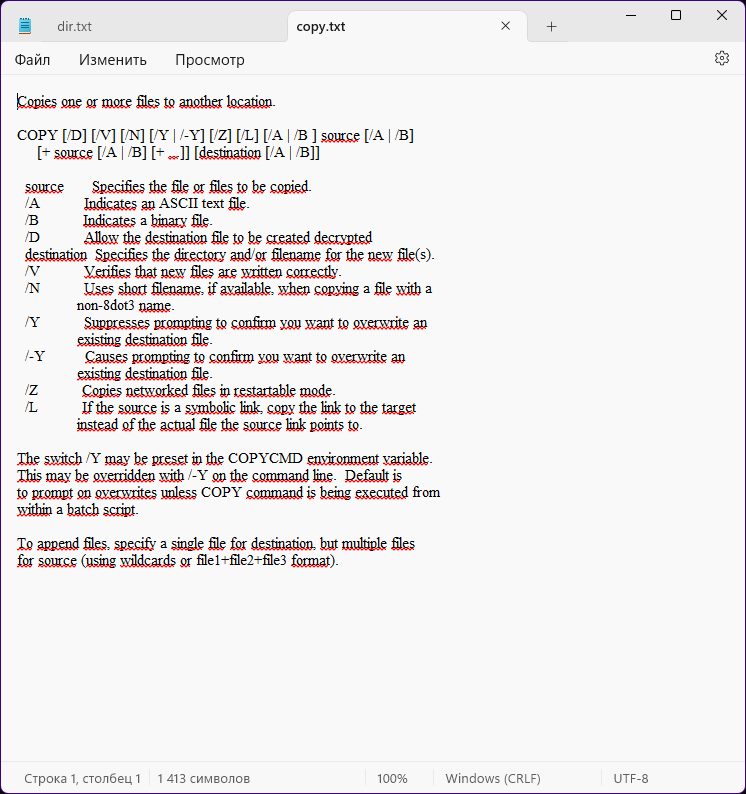
Задание 3. Создать текстовый файл, содержащий справочные материалы по командам DIR, COPY, XCOPY. Для этого вводим следующие команды:

1. DIR /?>dir.txt
2. COPY /?>copy.txt
3. XCOPY /?>xcopy.txt

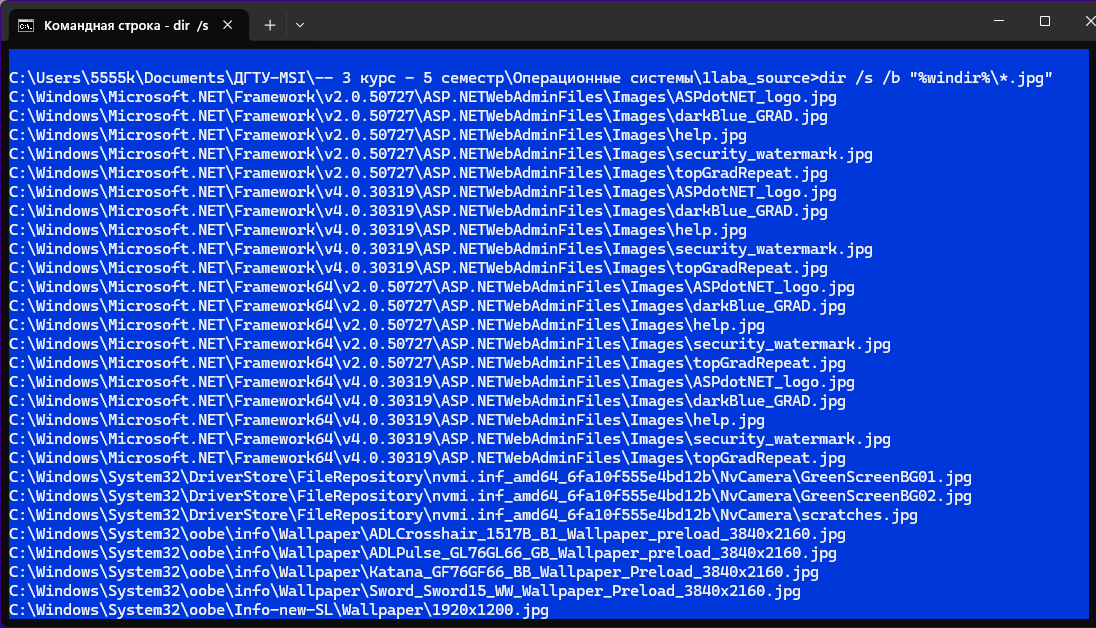


Результат можно просмотреть в программе Блокнот.

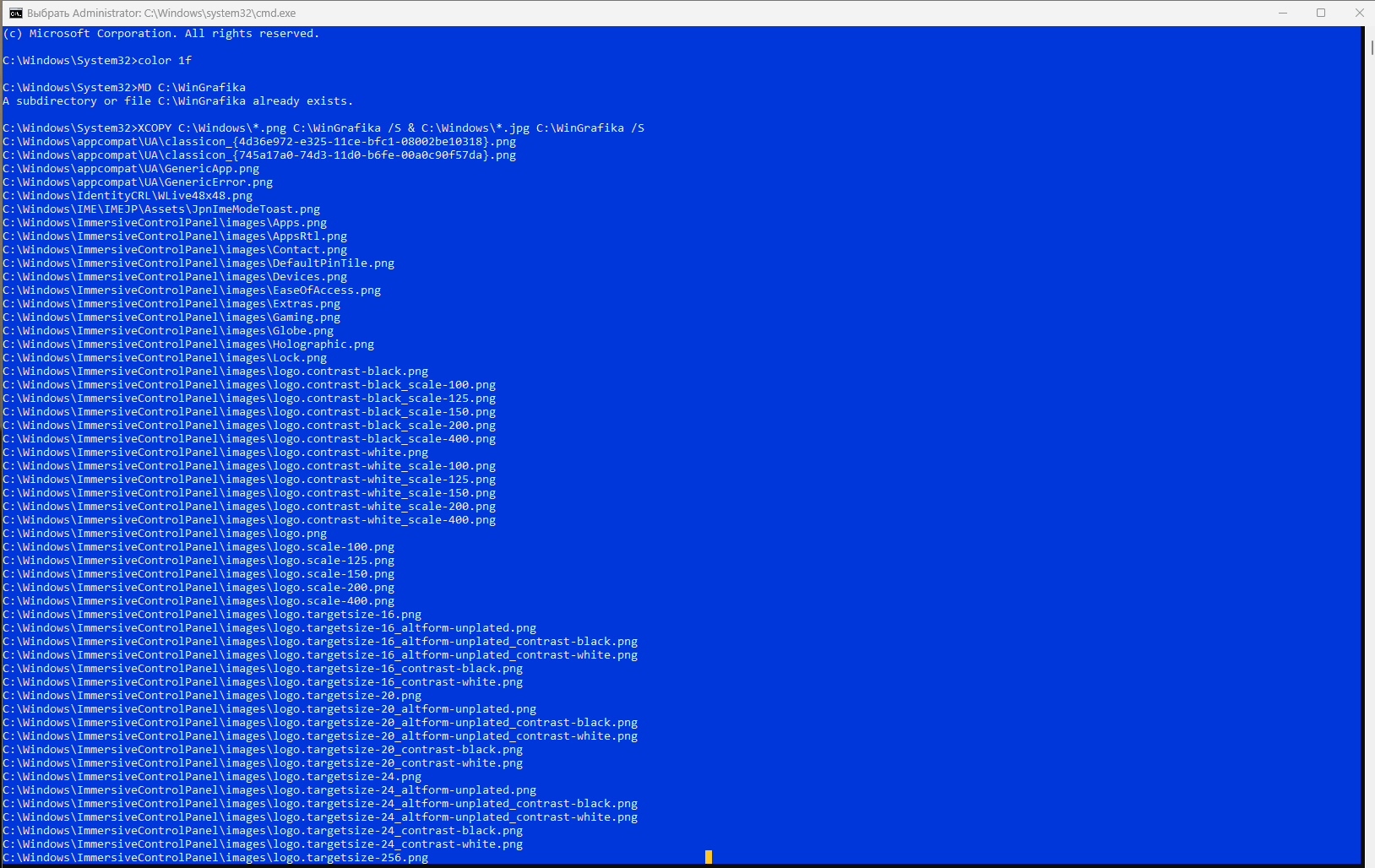




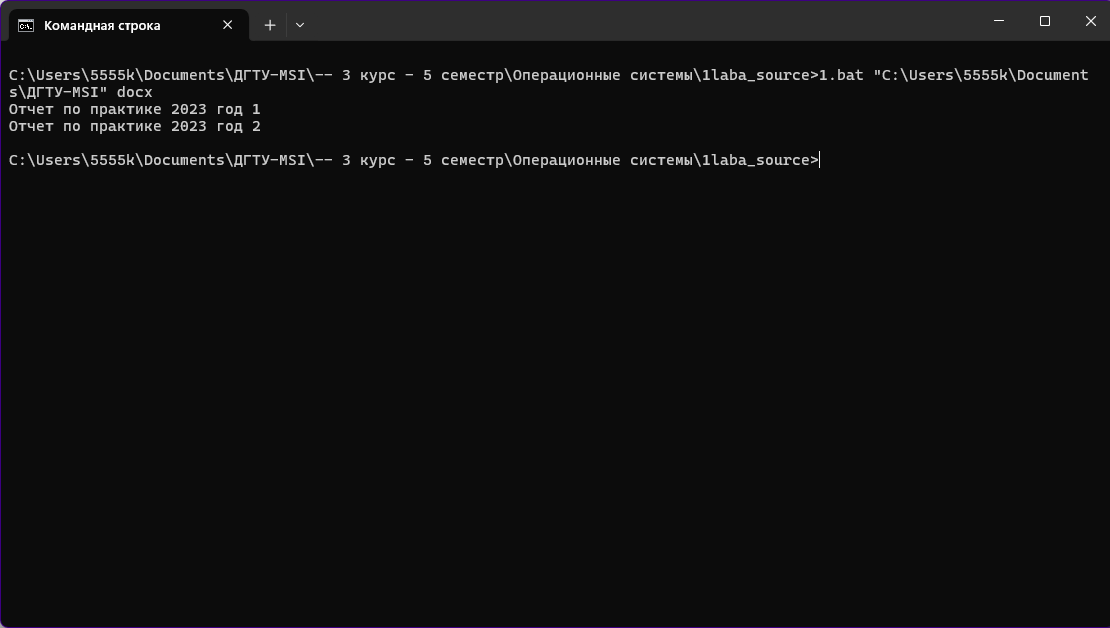
Задание 4. Вывести только файлы jpg из %Windows% и все подкаталогов по именам на экран и в файл. Для этого вводим следующую команду



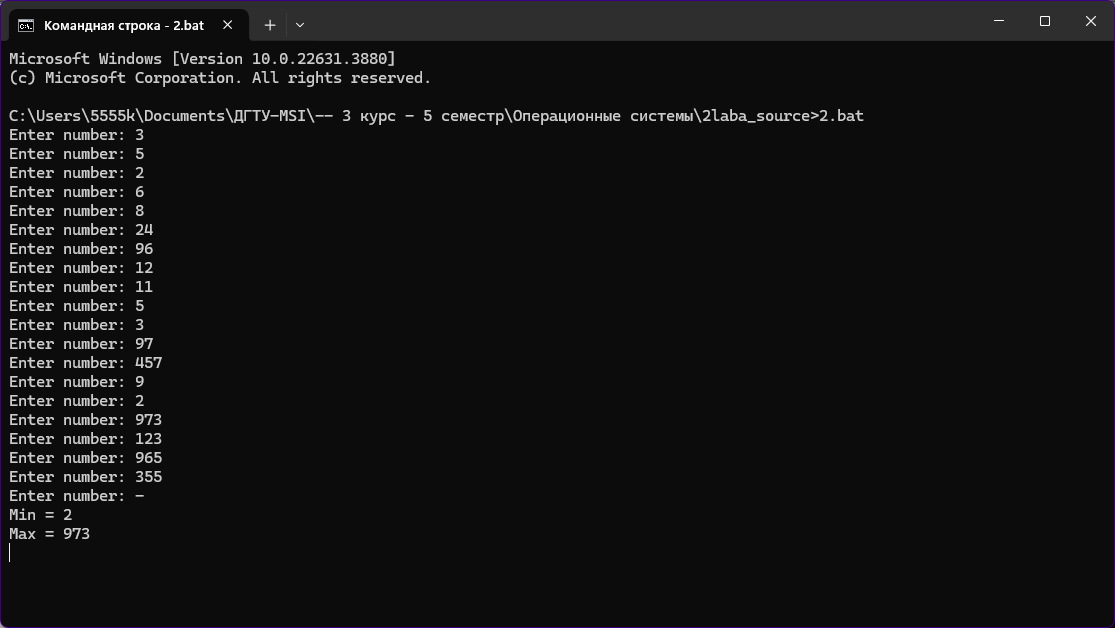
Задание 5. Скопировать все имеющиеся в каталоге Windows растровые графические файлы в каталог WinGrafika на диске С:. Если диск С: недоступен, использовать любой другой доступный диск. Растровые расширения \*.jpg \*.png



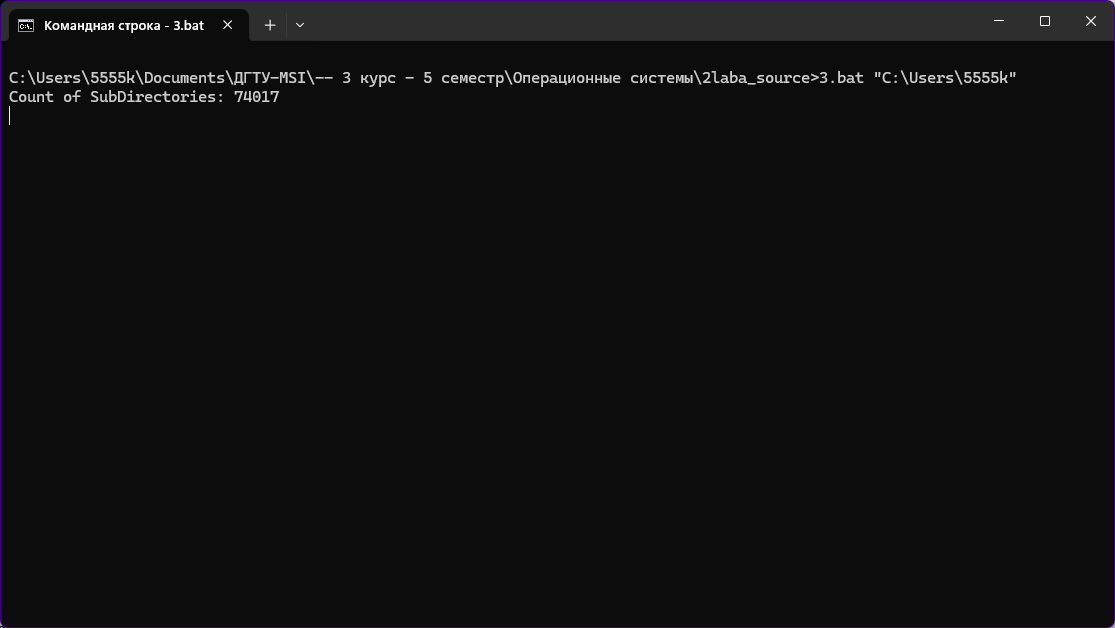
**Задание 6.1**. Вывод на экран имен всех файлов с указанным расширением, находящихся в каталоге, имя которого задается при запуске командного файла первым параметром. Расширение файлов задается вторым параметром.



Задание 6.2. Среди введенных с клавиатуры целых чисел (использовать SET /P) найти наибольшее и наименьшее. Признак конца ввода – знак «-».

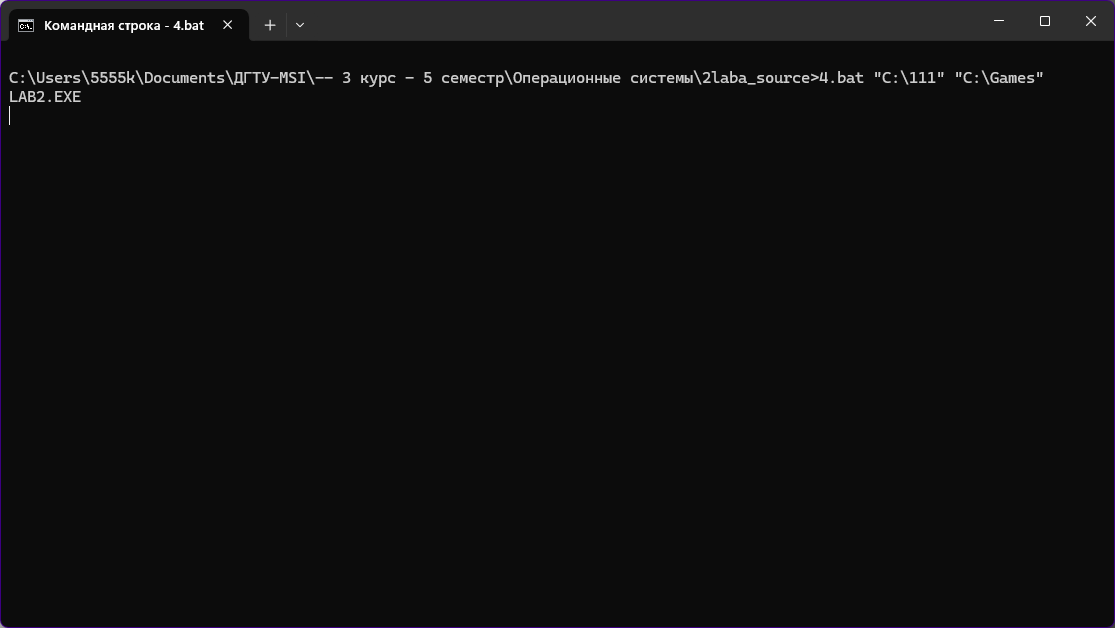


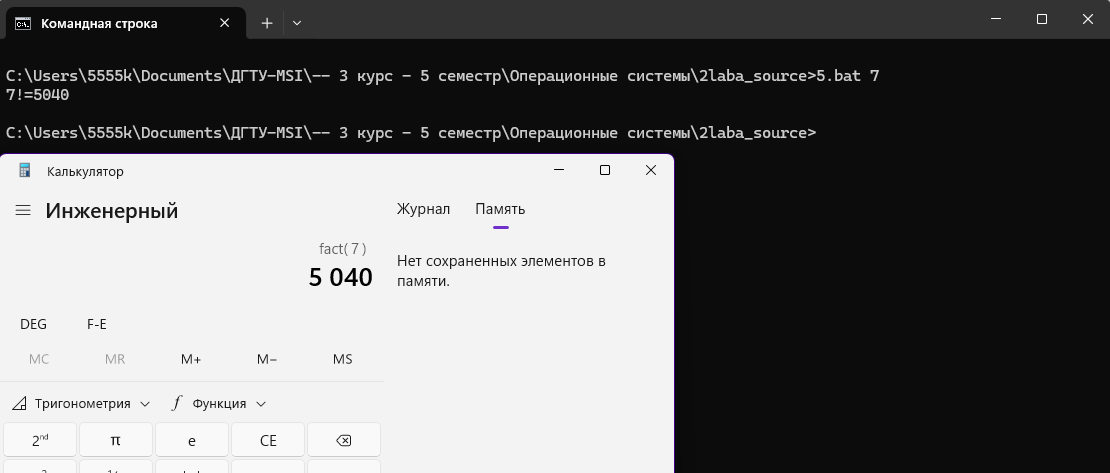
Задание 6.3 В заданном каталоге и его подкаталогах найти общее количество подкаталогов. На экран вывести только требуемый результат.



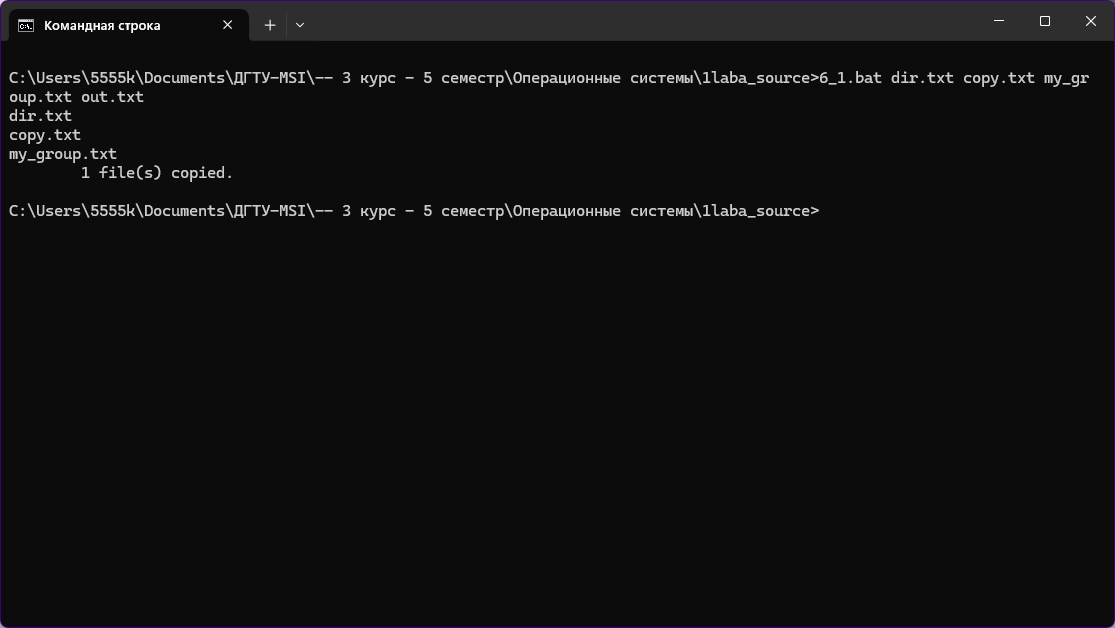
Задание 6.4. В каталогах, имена которых заданы первым и вторым параметрами командного файла, найти и вывести на экран имена файлов (расширения могут быть любые), присутствующие как в первом, так и во втором каталоге. Следует использовать только один оператор FOR.

|  |  |
| --- | --- |
| %~Ni | Из переменной %i выделяется только имя файла |
| %~Xi | Из переменной %i выделяется расширение имени файла |

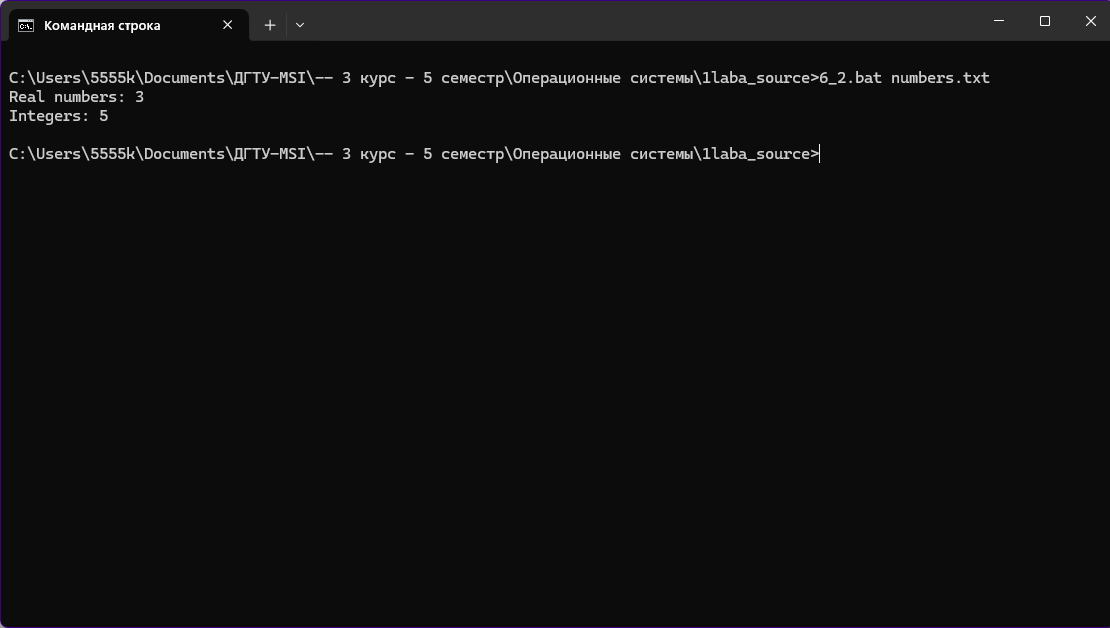


Задание 6.5. Вычисление и вывод на экран значения факториала целого числа, задаваемого при запуске КФ. Предусмотреть проверку заданного значения и при задании отрицательного значения или значения, превышающего максимально возможную величину, выводить соответствующие сообщения. Для проверки правильности вычислений использовать калькулятор. 

Задание 7.1. Проверка наличия трех текстовых файлов на диске и объединения их в один файл.



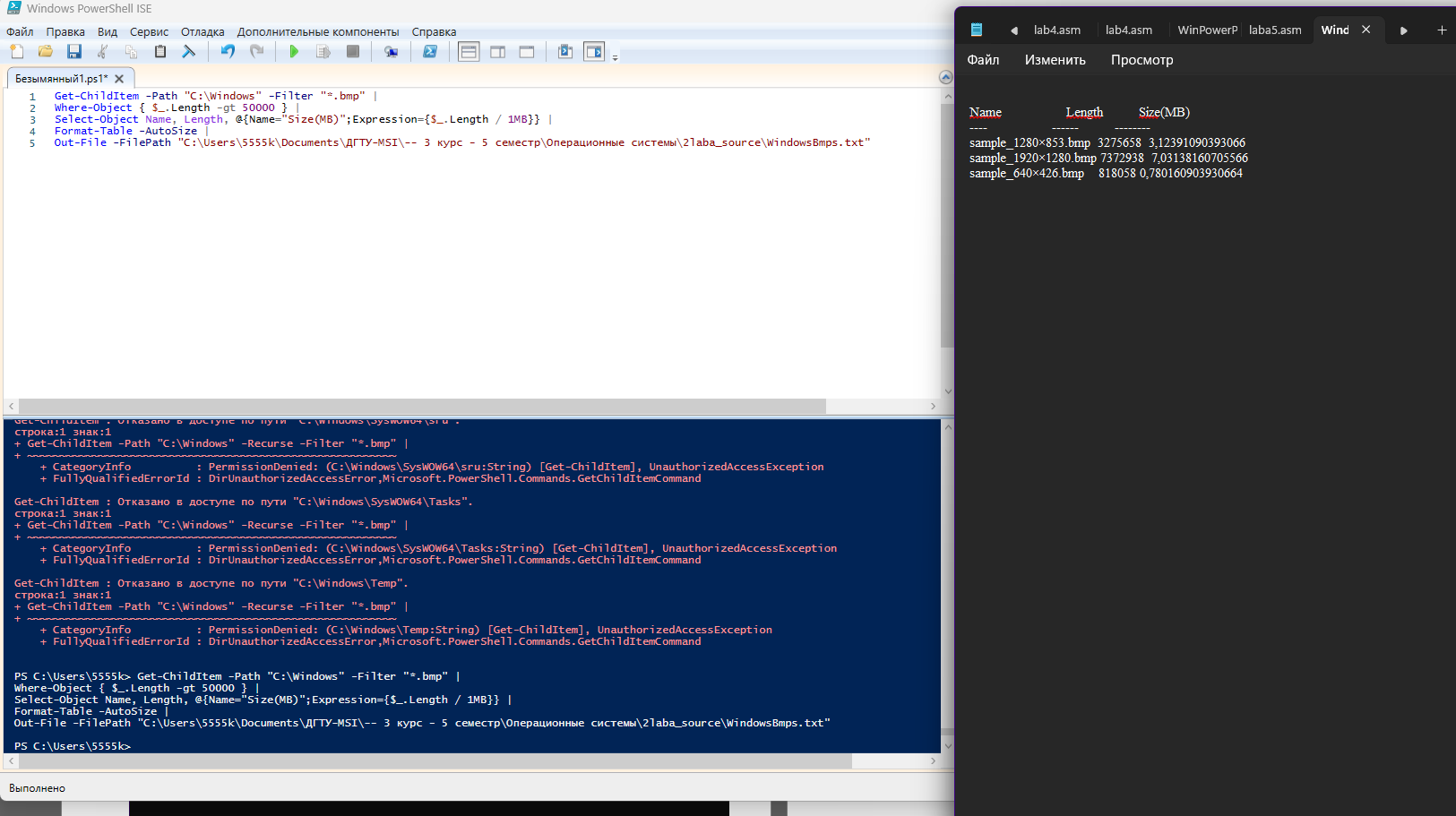
Задание 7.2. Подсчет количества вещественных чисел и целых чисел в текстовом файле. Вещественные и целые числа подсчитать отдельно. Считать, что слова в файле записаны в формате ОДНО СЛОВО В СТРОКЕ. Слово – это целое число (состоящее из десятичных цифр) или последовательность букв латинского алфавита (начинающаяся с буквы) или последовательность десятичных цифр с точкой (.) внутри строки. Имя файла задается первым параметром КФ.



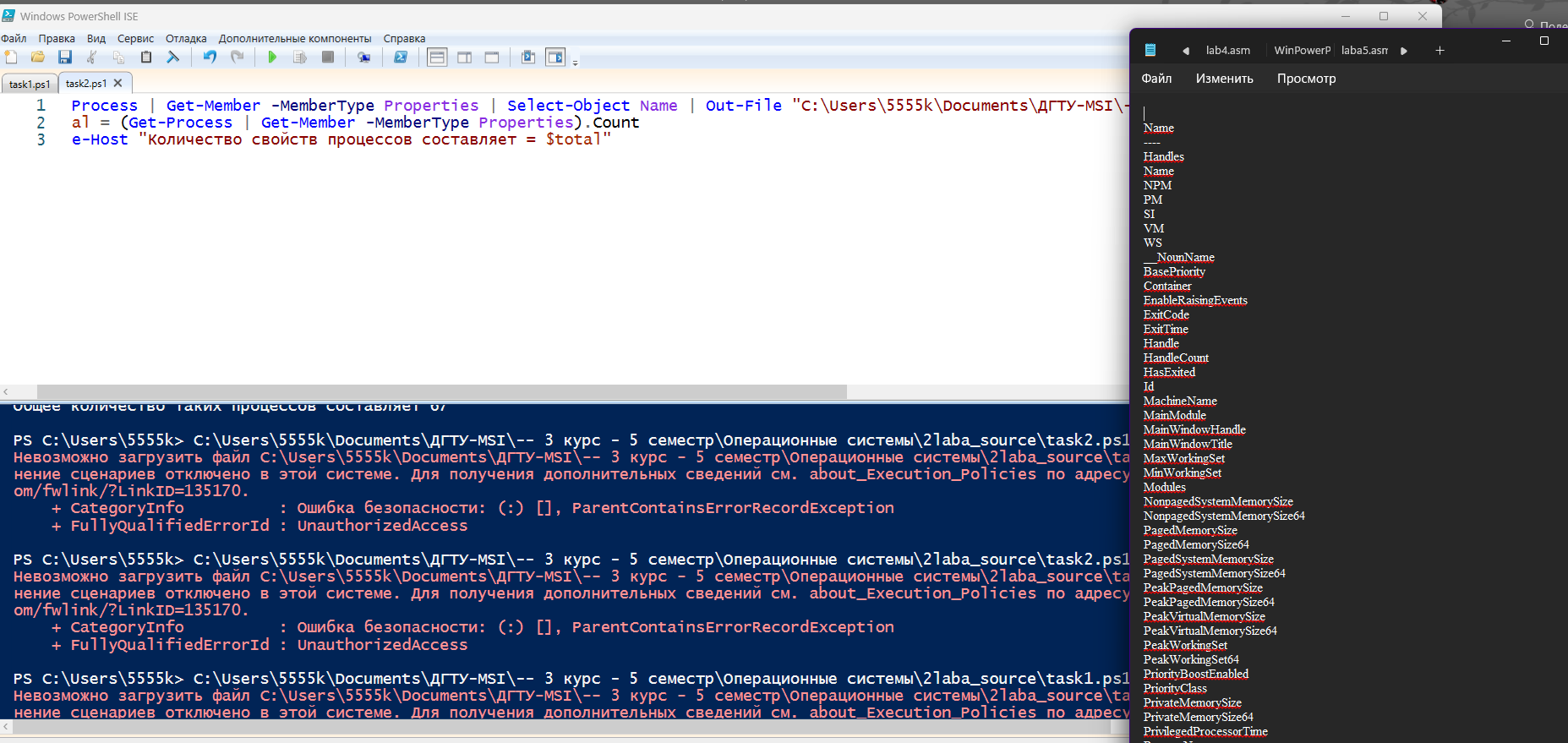
**ЛАБОРАТОРНАЯ РАБОТА №2 ОБОЛОЧКА КОМАНДНОЙ СТРОКИ WINDOWS POWERSHELL**

**Цель работы** – знакомство с основными возможностями оболочки командной строки Windows PowerShell.

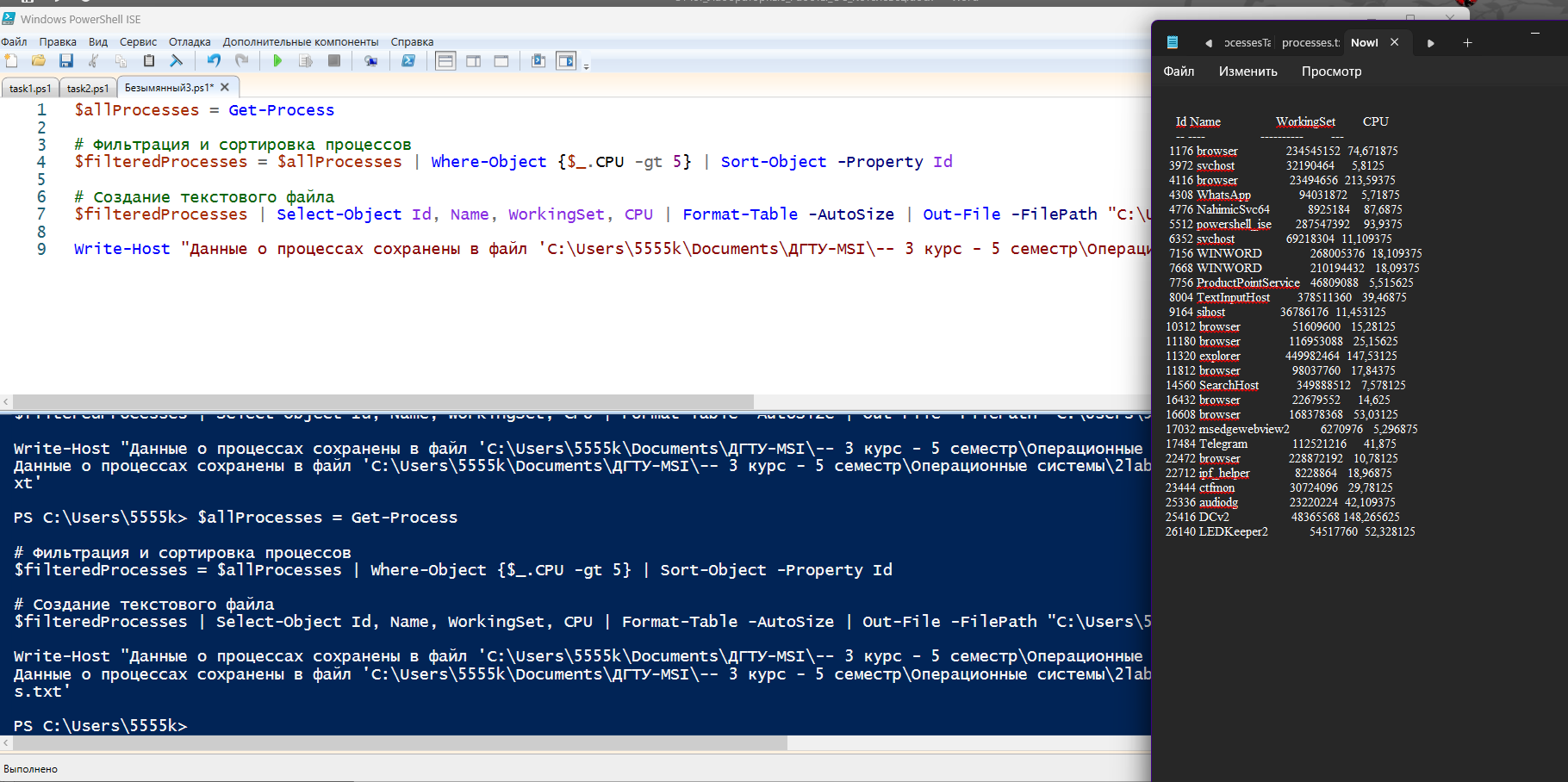
Задание 1. Вывести содержимое каталога Windows (для вариантов 5 и 10 – и подкаталогов) по указанному в табл. 4 формату на экран и в текстовый файл. У меня формат bmp, по размеру, где размер >50000



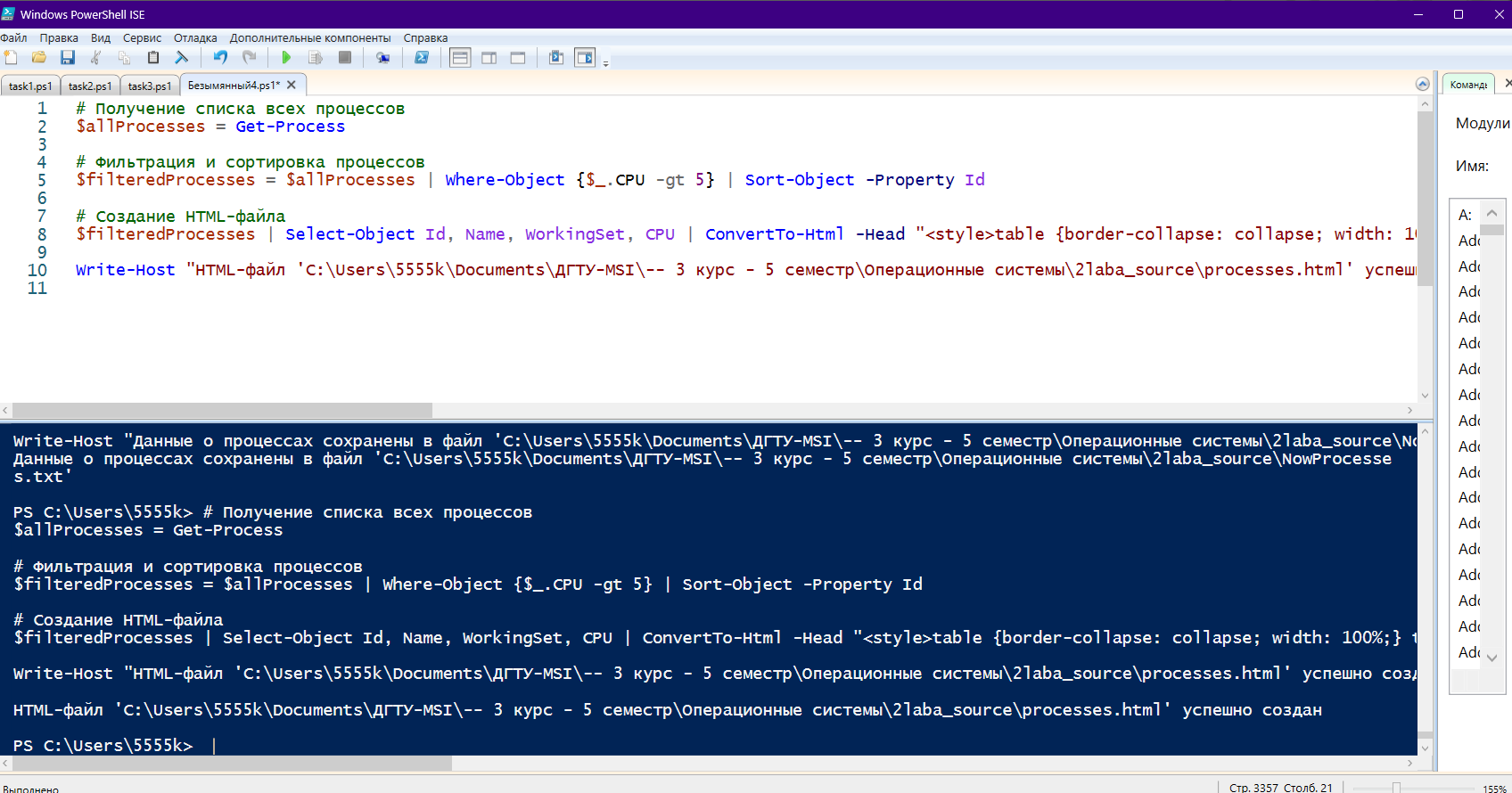
Задание 2. Вывести в текстовый файл список свойств процесса, возвращаемый командлетом Get-process и на экран – их общее количество.

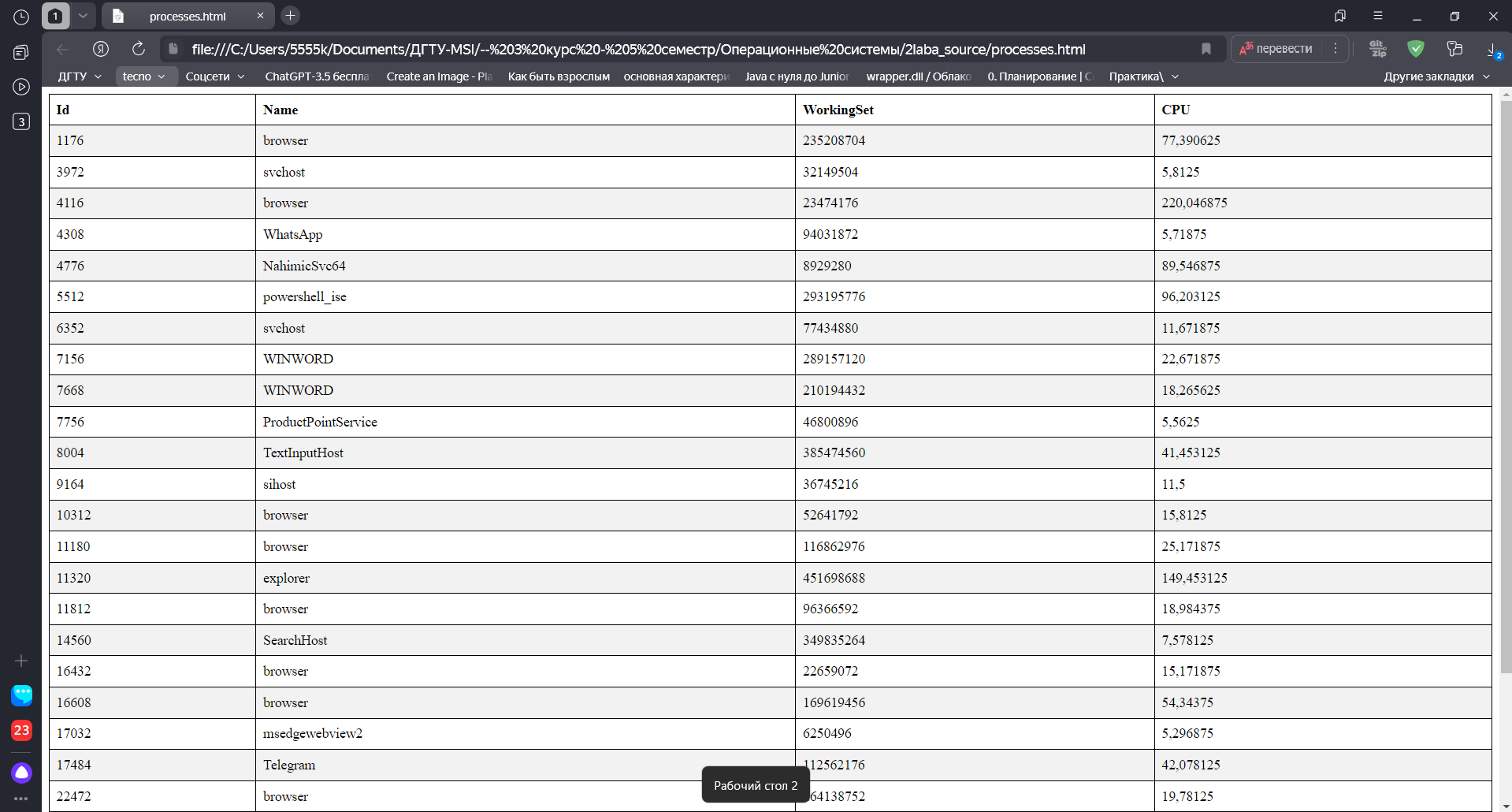


Задание 3. Cоздать текстовый файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в табл. 5 параметра. Имена параметров процессов указаны так же в табл. 5. У меня следующие выводимые параметры: Id, Имя процесса, WorkingSet, CPU – Сортировать по значению ID, при этом CPU > 5

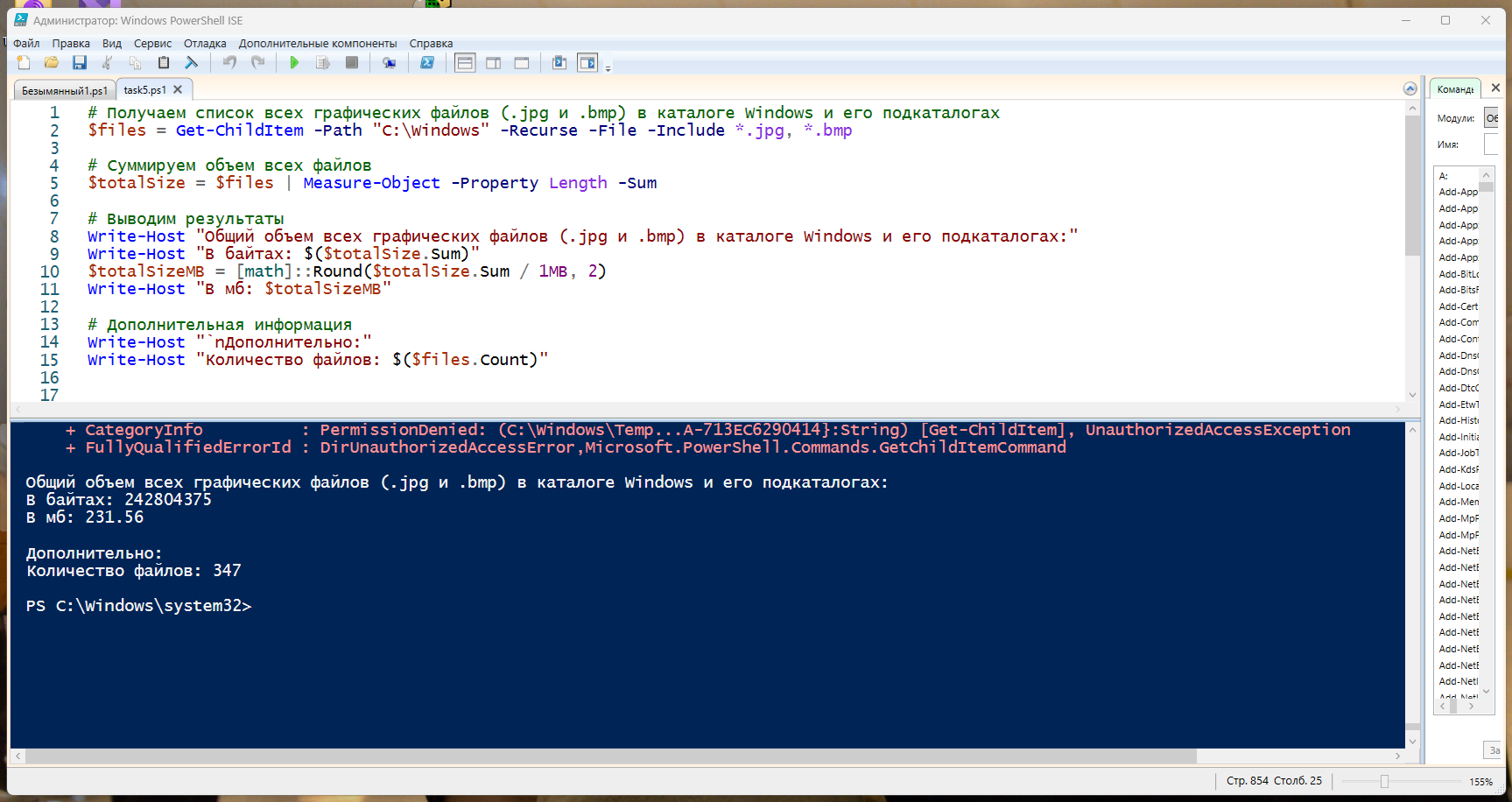


Задание 4. Cоздать HTML-файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в табл.5 параметра. Имена параметров процессов указаны в табл. 5. У меня следующие выводимые параметры: Id, Имя процесса, WorkingSet, CPU

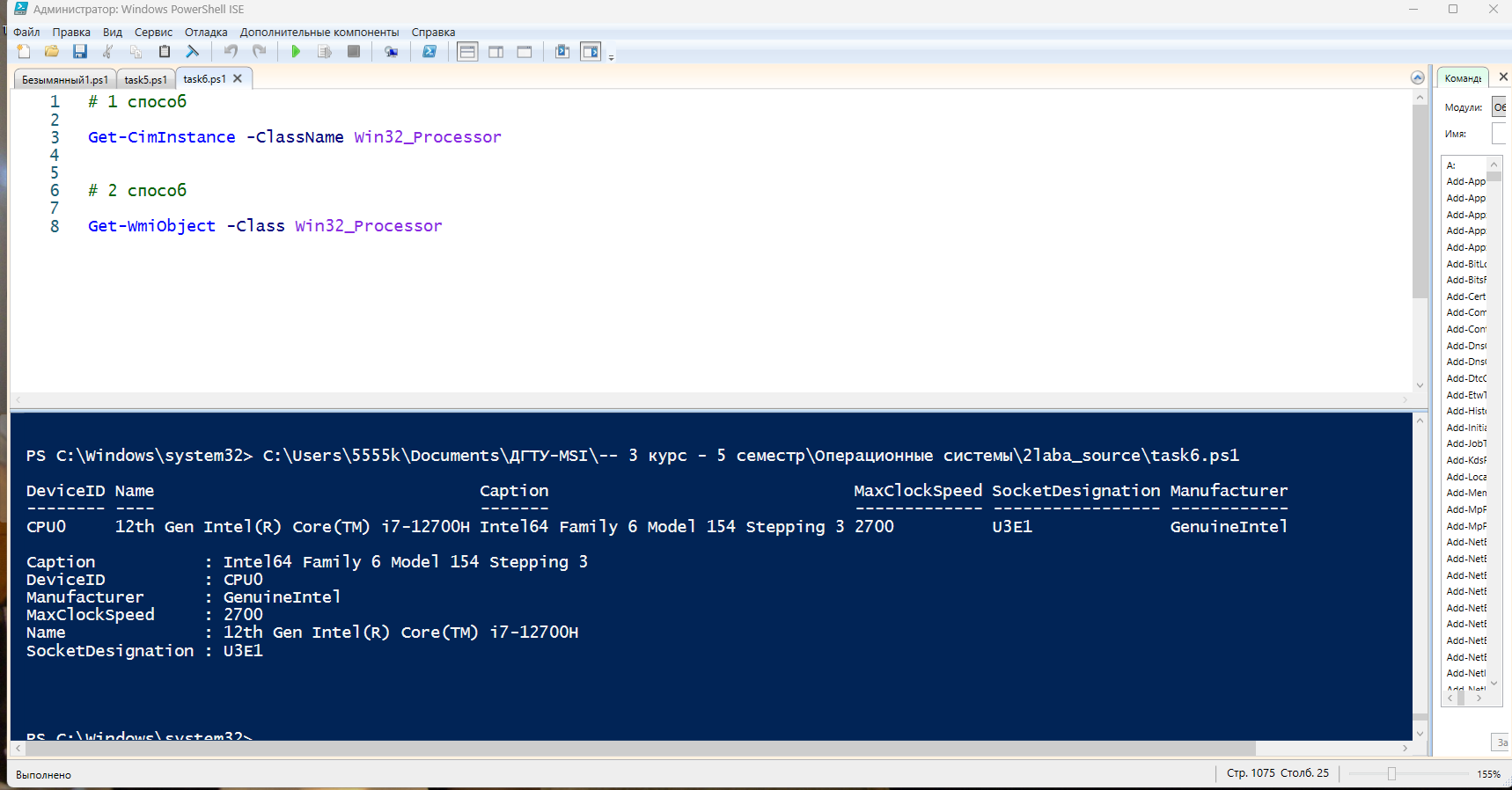




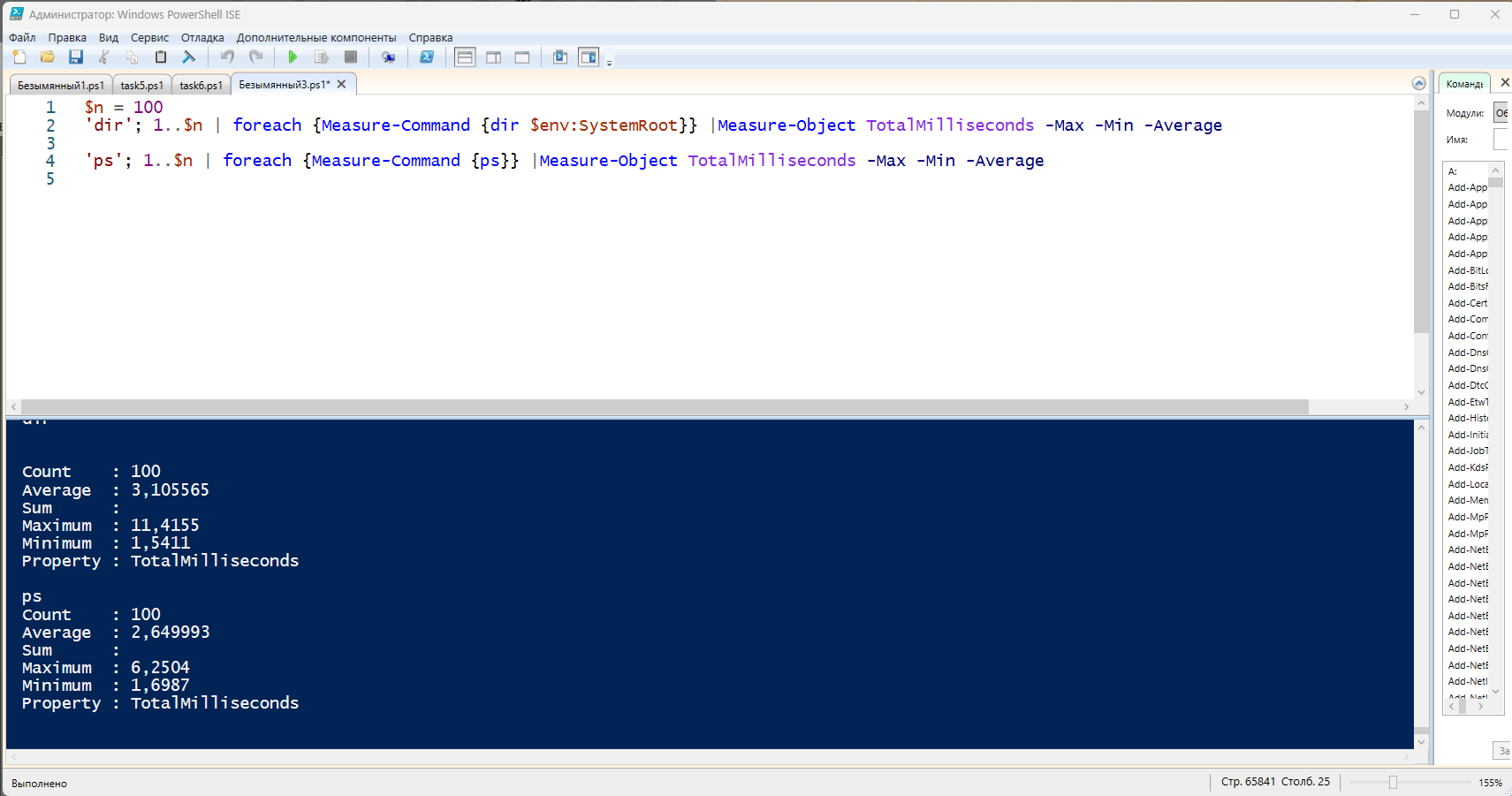
Задание 5. Найти суммарный объем всех графических файлов (bmp, jpg), находящихся в каталоге Windows и всех его подкаталогах.



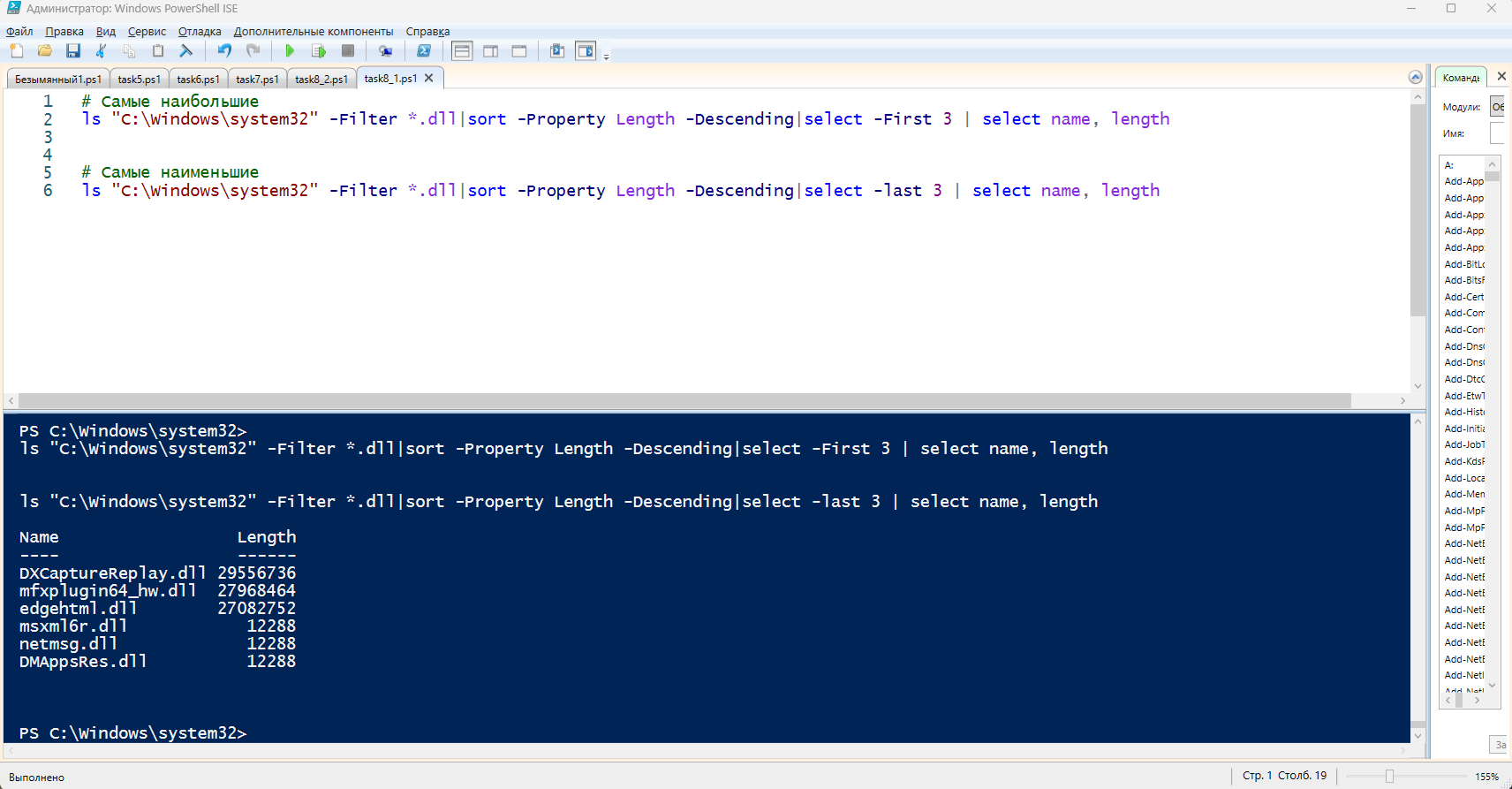
Задание 6. Вывести на экран сведения о ЦП компьютера



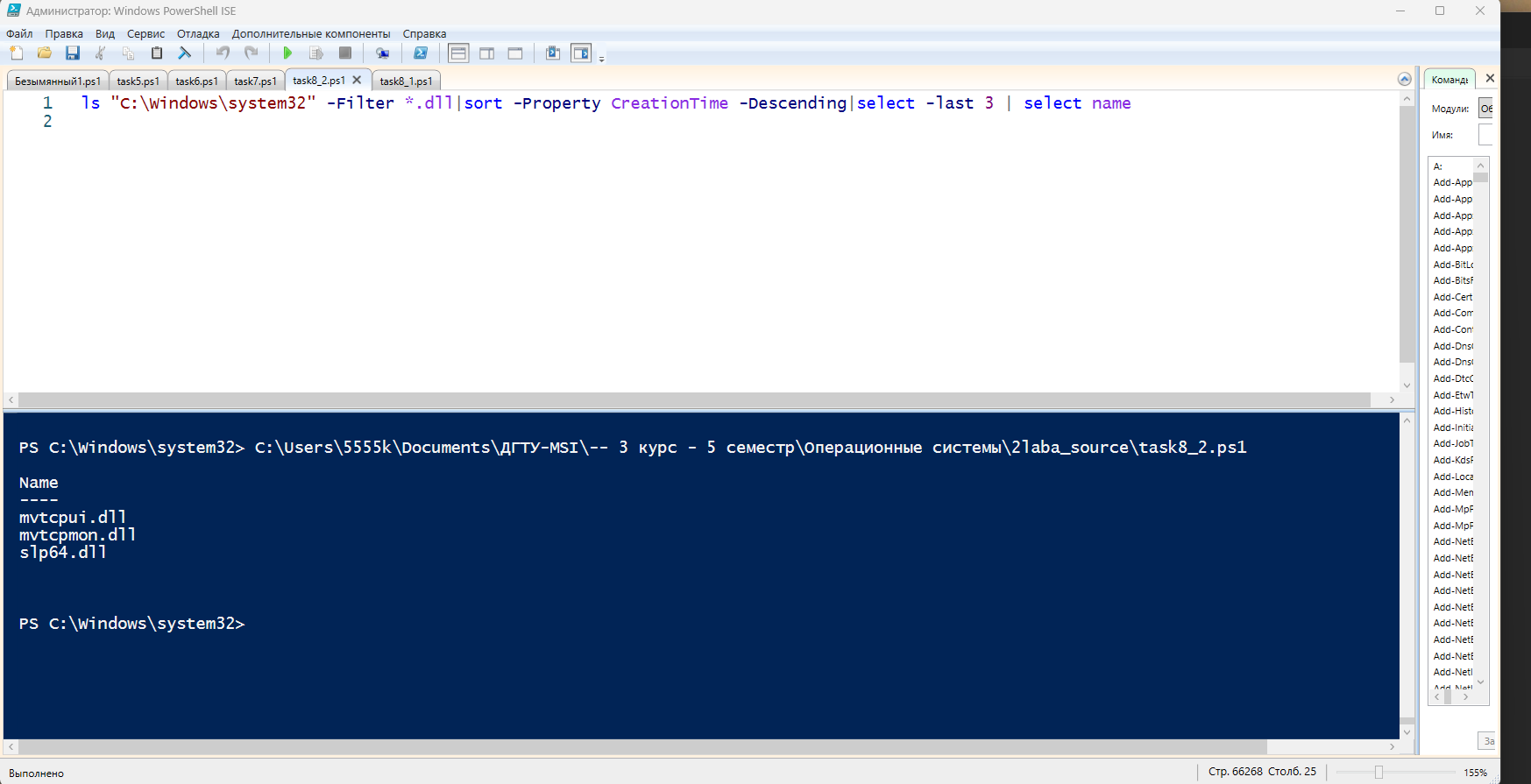
Задание 7. Найти максимальное, минимальное и среднее значение времени выполнение командлетов dir и ps



Задание 8.1. Нахождение в каталоге windows\system32 имен трех dll наибольшего и наименьшего размеров



Задание 8.2. Нахождение в каталоге windows\system32 имен трех dll с самой ранней датой создания.



Контрольные вопросы:

1. Типы команд PowerShell (PS).

PowerShell поддерживает следующие типы команд:

* Простые команды (simple commands)
* Скрипты (scripts)
* Функции (functions)
* Сценарии (scripts)
* Конвейеры (pipelines)

2. Имена и структура командлетов.

омандлеты в PowerShell имеют следующую структуру:

* Название командлета (например, Get-Process)
* Параметры с их значениями
* Возможные модификаторы (например, -Verbose)

3. Псевдонимы команд.

Псевдонимы в PowerShell используются для краткости и удобства. Например, вместо Get-Process можно использовать gp.

4. Просмотр структуры объектов.

Для просмотра структуры объекта используется cmdlet Get-Member.

5. Фильтрация объектов в конвейере. Блок сценария.

Фильтрация объектов в конвейере выполняется с помощью командлетов, таких как Where-Object или Select-Object.

Get-Process | Where-Object { $\_.CPU -gt 100 }

6. Какую информацию выводит команда Get-Help \* ?

Get-Help \* выводит справку по всем доступным командлетам в PowerShell.

7. Командлеты для форматирования выводимой информации.

Основные командлеты для форматирования:

* Format-Table
* Format-List
* Format-Custom

8. Перенаправление выводимой информации.

Вывод можно перенаправить в файл с помощью > или >> операторов.

9. Управляющие инструкции PS.

Управляющими инструкциями в PowerShell являются:

* if
* else
* switch
* foreach
* while
* do-while

10. Назначение регулярных выражений.

Регулярные выражения в PowerShell используются для поиска и замены паттернов в строках. В PS применяются с операторами -match, -replace.

11.Сохранение данных в текстовом файле и html-файле.

Данные можно сохранять в файл с помощью Out-File или Export-Csv для CSV файлов и ConvertTo-Html для HTML файлов.

Для сохранения данных в текстовый файл: Get-Process | Out-File "processes.txt"

Для HTML-файла: Get-Process | ConvertTo-Html | Out-File "processes.html"

12.Получение справочной информации в PS.

Справку можно получить с помощью команды Get-Help или ?.

13.Как создать массив в PS?

Массив создается с помощью синтаксиса @(). Пример: $arr = @(1, 2, 3)

14.Как объединить два массива?

Два массива можно объединить с помощью оператора +

$arr1 = @(1, 2) $arr2 = @(3, 4)

$combined = $arr1 + $arr2

15.Как увеличить размер созданного в PS массива?

Массивы в PS динамически расширяемы. Добавление новых элементов не требует дополнительных действий.Пример: $arr += 5

16.Как ввести данные в массив?

Можно использовать оператор += для добавления новых элементов:

$arr = @()

$arr += 1

17.Использование командлета Out-Null.

Out-Null используется для отбрасывания вывода без сохранения его в переменной.

18.Оператор PowerShell –match.

Оператор -match используется для проверки соответствия строки регулярному выражению.

Пример: "abc123" -match "\d+" # true

19.Использование символа ^ в командлетах.

Символ ^ в PowerShell используется как отрицательный модификатор.

20.Использование символа $ в командлетах.

Символ $ в PowerShell используется для указания переменных.

21.Количественные модификаторы (квантификаторы).

Квантификаторы в регулярных выражениях используются для определения количества повторений.

\* — ноль или более,

+ — одно или более,

? — ноль или одно.

22.Использование групп захвата.

В регулярных выражениях PS группы захвата заключаются в скобки (). Пример:

"abc1 23" -match "(abc)(\d+)"

$matches[1] # "abc"

$matches[2] # "123"

23.Командлеты для измерения свойств объектов.

Основные командлеты для измерения свойств объектов:

Для измерения свойств используются командлеты:

Measure-Object — для измерения количества, суммы, среднего значения и других статистик объектов.