

1-3 выполнены.

Задание 4. Вариант 1. Вывести названия файлов больше 10000 байт из каталога Windows на экран и текстовый файл.

```
PS C:\OSST> Get-ChildItem "C:\Windows" -File | Where-Object {$_.Length -gt 10000} | Sort-Object
Length -Descending | Format-Table Name, Length -AutoSize | Out-File "C:\OSST\Лаба3\files.txt"
PS C:\OSST> Get-Content "C:\OSST\Лаба3\files.txt"
```

Name	Length
explorer.exe	6087496
unins000.exe	1207319
unins001.exe	1199175
HelpPane.exe	1065984
regedit.exe	370176
WMSysPr9.prx	316640
NSudo.exe	252928
notepad.exe	200704
splwow64.exe	164352
Windows.exe	140800
PFR0.log	129440
bfsvc.exe	93696
twain_32.dll	76800
bootstat.dat	67584
fidelity.exe	45568
mib.bin	43131
DirectX.log	35934
Professional.xml	30831
unins001.dat	21735
hh.exe	18432
winhlp32.exe	11776
write.exe	11264
unins000.dat	10830
Info.xml	10772

В коде я указала что мне нужны файлы в каталоге Windows размером больше 10000 байт, сортированные по размеру, а формат таблицы для вывода должен включать название и размер. Сохранить всё в файле files.txt и вывести всё на экран уже второй командой.

Задание 5. Вывести список свойств Get-process и их количество.

```
PS C:\OSST> Get-Process | Out-File "C:\OSST\Лаба3\process_list.txt"; (Get-Process).Count
171
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
159	9	7904	6788	178,67	4340	0	AggregatorHost
316	14	23624	38780	0,61	19352	26	ai
129	8	2068	1228	0,23	1684	0	amdfendrsr
176	11	6396	12184	0,23	21208	0	audiodg
267	10	6852	9028	0,08	2640	26	chrome
746	54	281076	160400	283,66	7380	26	chrome
212	19	18336	21024	1,33	8512	26	chrome
288	22	45616	83444	4,66	10256	26	chrome
284	20	23872	55156	0,41	12884	26	chrome
228	14	14576	19256	2,00	14564	26	chrome
328	23	47128	93944	8,77	14772	26	chrome

Задание 6. Вариант 9. Создать текстовый файл, содержащий список процессов.

```
PS C:\OSST> Get-Process | Where-Object {$_.CPU -gt 5} | Sort-Object Id | Select-Object Id, Name, WorkingSet, CPU | Out-File "C:\OSST\Лаба3\processes.txt"
```

Получаем все процессы, фильтруем процессы с CPU > 5, сортируем по Id, выбираем указанные параметры и сохраняем в текстовый файл.

Id	Name	WorkingSet	CPU
4	System	28672	29820,953125
100	Registry	65925120	23,046875
436	svchost	7823360	798,765625
496	csrss	2461696	26,328125
660	services	7979008	337,640625
668	lsass	26947584	426,5
864	svchost	46616576	1394,4375
1000	svchost	20914176	2818,546875
1124	svchost	6209536	20,4375
1152	svchost	5898240	7,203125
1196	svchost	12832768	377,953125
1252	sihost	29872128	15,703125
1268	svchost	11964416	80,75
1304	svchost	10031104	12,5
1372	svchost	8888320	171,265625
1400	svchost	9953280	87,796875
1436	svchost	7434240	22,90625
1452	svchost	4509696	351,421875
1608	svchost	9089024	215,234375
1692	svchost	7983104	610,75
1768	svchost	56946688	8821,671875
1776	svchost	2101248	14,09375
1868	Memory Compression	77856768	7085,453125
1912	svchost	3747840	12,546875

Задание 7. Создание HTML файла со списком файлов по 5-ой таблице.

```
PS C:\OSST> Get-ChildItem "C:\Windows" -File | Where-Object {$_.Length -gt 10000} | Sort-Object Length -Descending | ConvertTo-Html -Property Name, Length, CreationTime, Attributes | Out-File "C:\OSST\Лаба3\files.html"
```

Получает файлы из каталога Windows, фильтруем по размеру, сортируем также по размеру, преобразуем в HTML формат и сохраняем

Name	Length	CreationTime	Attributes
explorer.exe	6087496	17.07.2025 7:05:40	Archive
unins000.exe	1207319	25.11.2024 12:02:35	Archive
unins001.exe	1199175	25.11.2024 12:02:39	Archive
HelpPane.exe	1065984	17.07.2025 7:09:05	Archive
regedit.exe	370176	17.07.2025 7:09:22	Archive
WMSysPr9.prx	316640	07.12.2019 12:10:11	Archive
NSudo.exe	252928	09.05.2022 8:10:00	Archive
notepad.exe	200704	17.07.2025 7:10:42	Archive
splwow64.exe	164352	17.07.2025 7:05:34	Archive
Windows.exe	140800	09.05.2022 22:17:21	Archive
PFRO.log	129440	24.11.2024 21:46:15	Archive
bfsvc.exe	93696	17.07.2025 7:05:31	Archive
twain_32.dll	76800	17.07.2025 7:09:41	Archive
bootstat.dat	67584	17.07.2025 7:19:30	System, Archive
fidelity.exe	45568	18.04.2022 1:30:33	Archive
mib.bin	43131	07.12.2019 12:08:58	Archive
DirectX.log	35934	03.01.2025 22:14:59	Archive
Professional.xml	30831	07.12.2019 17:37:58	Archive
unins001.dat	21735	25.11.2024 12:02:39	Archive
hh.exe	18432	07.12.2019 12:09:39	Archive
winhlp32.exe	11776	07.12.2019 12:10:00	Archive
write.exe	11264	07.12.2019 17:36:46	Archive
unins000.dat	10830	25.11.2024 12:02:34	Archive
Info.xml	10772	13.02.2025 10:47:46	Archive

Задание 8. Найти общий размер всех графических файлов в Windows и подкаталогах.

```
(Get-ChildItem "C:\Windows" -Recurse -Include "*.bmp","*.jpg" | Measure-Object -Property Length -Sum).
Sum
Get-ChildItem : Отказано в доступе по пути "C:\Windows\CSC".
строка:1 знак:2
+ (Get-ChildItem "C:\Windows" -Recurse -Include "*.bmp","*.jpg" | Measu ...
+ ~~~~~
+ CategoryInfo          : PermissionDenied: (C:\Windows\CSC:String) [Get-ChildItem], Unauth
  orizedAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChil
  dItemCommand

Get-ChildItem : Отказано в доступе по пути "C:\Windows\System32\LogFiles\WMI\RtBackup".
строка:1 знак:2
+ (Get-ChildItem "C:\Windows" -Recurse -Include "*.bmp","*.jpg" | Measu ...
+ ~~~~~
+ CategoryInfo          : PermissionDenied: (C:\Windows\Syst...es\WMI\RtBackup:String) [Ge
  t-ChildItem], UnauthorizedAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChil
  dItemCommand

50921040
```

Сначала он немного поругался что нет доступа, но остальное всё же посчитал.

Задание 9. Сведения о ЦП компьютера.

```
PS C:\OSST> Get-CimInstance Win32_Processor | Select-Object Name, Manufacturer,
NumberOfCores, NumberOfLogicalProcessors, MaxClockSpeed, CurrentClockSpeed

Name                                     : AMD PRO A8-8650B R7, 10 Compute Cores 4C+6G
Manufacturer                           : AuthenticAMD
NumberOfCores                           : 2
NumberOfLogicalProcessors               : 4
MaxClockSpeed                           : 3200
CurrentClockSpeed                       : 3200
```

Задание 10. Измерить максимальное и минимальное время выполнения команд.

```
PS C:\OSST> $dirTimes = @()
>> $psTimes = @()
>> for ($i = 0; $i -lt 10; $i++) {
>>     $dirTimes += (Measure-Command { dir }).TotalMilliseconds
>>     $psTimes += (Measure-Command { ps }).TotalMilliseconds
>> }
>> $maxDirTime = ($dirTimes | Measure-Object -Maximum).Maximum
>> $minDirTime = ($dirTimes | Measure-Object -Minimum).Minimum
>> $avgDirTime = ($dirTimes | Measure-Object -Average).Average
>> $maxPsTime = ($psTimes | Measure-Object -Maximum).Maximum
>> $minPsTime = ($psTimes | Measure-Object -Minimum).Minimum
>> $avgPsTime = ($psTimes | Measure-Object -Average).Average
>> "[DIR]: MAX: $maxDirTime ms, MIN: $minDirTime ms, AVG: $avgDirTime ms"
>> "[PS]: MAX: $maxPsTime ms, MIN: $minPsTime ms, AVG: $avgPsTime ms"
[DIR]: MAX: 1.4781 ms, MIN: 1.0427 ms, AVG: 1.17975 ms
[PS]: MAX: 13.5415 ms, MIN: 5.0844 ms, AVG: 6.97947 ms
```

Измеряем время выполнения команды dir и ps по 10 раз. Вычисляем максимальное, минимальное и среднее значение времени выполнения.

Задание 11. Вариант 3. Посчитать количество положительных и чётных чисел.

```
PS C:\OSST\Лаба3> (Get-Content "nn.txt" | Where-Object {$_. -gt 0}).Count;
(Get-Content "nn.txt" | Where-Object {$_. % 2 -eq 0}).Count
49
29
```

Текст у меня с рандомными числами.

```
-23 67 -15 88 0 -48 32 99 -87 14 -6 55 -31 70 -92 43 -29 81 -4 60 -76 22 -1 45 -54 30 78 16 -68 49 -20 12 73 -38 6 -59 26 85 -33
```

Батник для привязки к Git

```
@echo off
```

```
git config --global user.name "Victoria"
git config --global user.email "vikaburdejnaa4@gmail.com"
echo Git configured successfully!
pause
git config --global --unset user.name
git config --global --unset user.email
```

При запуске батника происходит подключение к моему профилю. Когда закрываю, программа автоматически отвязывает мой профиль.