

Всі конкретні значення (назва файлу, розширення, кількість і тд) слід передавати аргументами (мати можливість вказати).

каталог == папка == директорія

1. У заданому каталозі DIR залишити перші файли, які не перевищують сумарно заданої кількості V байтів, а решту перемістити у заданий каталог DIR_TO.	2. З усіх підкаталогів заданого каталогу DIR перекопіювати у заданий каталог DIR_TO усі файли, які мають задане розширення назви EXT (наприклад .cpp).
3. У заданому каталозі DIR залишити перші N файлів, які не перевищують заданої кількості V байтів, а решту перемістити у заданий каталог DIR_TO.	4. В усіх підкаталогах заданого каталогу DIR знищити усі файли, які мають задане розширення назви (наприклад “.xxx”).
5. У заданому каталозі DIR залишити файли, які за часом модифікації “свіжіші” файлів з такою ж назвою у заданому каталозі DIR_DIFF, або не зустрічаються в DIR_DIFF.	6. В усіх підкаталогах заданого каталогу DIR знищити усі файли, час зміни яких менший заданого t.
7. У заданому каталозі DIR усі файли(крім каталогів) перемістити у підкаталоги за назвою розширення.	8. З усіх підкаталогів заданого каталогу DIR знайти підкаталог, який найдавніше модифікований.
9. У заданому каталозі DIR усі файли(крім каталогів) посортувати у підкаталоги за назвою до першої крапки зліва.	10. У заданому каталозі DIR створити підкаталоги з назвами, заданими у файлі FILE.
11. У заданому каталозі DIR створити таку ж підсистему каталогів, що і в каталозі DIR_FROM.	12. Знайти сумарний об’єм файлів у підкаталозі заданого каталогу DIR, який має найбільше звичайних файлів.
13. У заданому каталозі DIR знищити вміст підкаталогів із заданою назвою n і повідомити їх сумарний об’єм.	14. У заданому каталозі DIR з усіх підкаталогів з заданою назвою SUB_DIR_NAME, вміст (звичайні файли) перемістити у заданий каталог DIR_TO_MOVE_TO і повідомити їх сумарний об’єм.
15. У заданому каталозі DIR знайти підкаталог з найбільшою кількістю підкаталогів.	16. Із заданого каталогу DIR перекопіювати перші N найбільших файлів у заданий каталог DIR_TO.
17. Заданий каталог DIR доповнити підкаталогами з їхнім вмістим (лише тими, яких у ньому не було) іншого заданого каталогу DIR_FROM.	18. Із заданого каталогу DIR вилучити підкаталоги, які є в іншому заданому каталозі DIR_TO_COMPARE.
19. У заданому каталозі DIR серед файлів, які мають однакові назви (відрізняються лише розширенням назви) знищити всі файли крім тих, які мають задане розширення (Наприклад “.xxx”)	20. Із заданого каталогу DIR перекопіювати N найменших файлів у заданий каталог DIR_TO.

21. Прочитати переданий файл (напр. file.txt), переконвертувати всі букви в нижній регістр, замінити пропуски на знаки нового рядка, посортувати результат, порахувати скільки слів що

мають повтори, і потім посортувати знову у зворотньому порядку (виводячи найбільш уживані слова спочатку), і потім відобразити 10 верхніх записів.

22. Знайдіть і видрукуйте шляхи до файлів у заданому каталозі DIR і його підкаталогах які були змінені X днів тому, потім видаліть ці файли

23. Зробіть простий шифр підстановки. На вхід команда повинна отримувати файл з перестановкою алфавіту якою буде замінюватись кожна буква тексту з файлу для шифрування. Також передбачити можливість дешифрування.

24. Із заданого каталогу t перенести перші n найменших файлів у заданий каталог z.

25. Із заданого каталогу DIR і його підкаталогів перекопіювати в заданий каталог DIR_TO ті каталоги, в яких всі файли модифіковані раніше за час X

26. В заданому каталозі DIR і його підкаталогах видаліть пропуски із всіх назв файлів і замініть їх на підкреслення

27. Зробіть пошук файлів з заданим розширенням (наприклад xml) які перевищують X kB (кілобайтів) у заданому каталозі DIR і його підкаталогах, і видрукуйте знайдені файли, їх кількість і дату їхньої останньої зміни

28. Знайдіть файл у вказаному каталозі DIR або його підкаталогах, у якому найбільша кількість слів та файл у якому найбільша кількість рядків. Шляхи до таких файлів видрукувати у окремому файлі.

29. Видалити всі символи окрім цифр із файлу test1.txt, потім зберегти це у файл test2.txt

30. Присвоїти вказаному каталозі DIR і його підкаталогам права на запис для поточного користувача. Видаліть файли у вказаному каталозі DIR і його підкаталогах показуючи підказку про видалення(Y/N?).

31. Стисніть та заархівуйте всі вихідні файли мови C (.h/.c) в каталозі DIR і його підкаталогах. Оновіть цей архів двома файлами foo.c bar.c. Видобудьте вміст архіву в окрему папку.

32. Заархівуйте(не стискаючи) всі вихідні з певним розширенням (наприклад, файли мови C# (.cs)) в каталозі DIR і його підкаталогах і стисніть створений таким чином tarball файл. Тепер видобудьте вміст архіву в окрему папку.

33. Знайдіть файл у вказаному каталозі DIR або його підкаталогах, у якому найбільша кількість рядків та файл у якому найбільша кількість символів. Шляхи до таких файлів видрукувати у окремому файлі.

34. Проаналізувати файли у заданому каталозі і його підкаталогах. Вивести список файлів, які змінювалися протягом останньої години. Посортувати всі файли по даті, вивести 'настаріші' 10 файлів. Вивести сумарний об'єм 10 найстаріших файлів. Скопіювати 10 найновіших файлів в каталог backup (може задаватися аргументом).

35. Знайдіть 100 найбільш уживаних слів у тексті про Франкенштейна що знаходиться за адресою <http://www.cs.cornell.edu/courses/cs2043/2012sp/frankenstein.txt>

36. Знайдіть 100 найбільш уживаних слів в тексті про Франкенштейна що знаходиться за адресою <http://www.cs.cornell.edu/courses/cs2043/2012sp/frankenstein.txt> які не є прийменниками, список прийменників можна знайти за адресою <http://www.cs.cornell.edu/courses/cs2043/2012sp/prepositions.txt> або <http://umich.edu/~umfandsf/other/ebooks/frank10.txt> <https://www.scrapmaker.com/data/wordlists/language/prepositions.txt>

37. Проаналізувати HTML текст (заув. HTML текст - це все що не є тегом), що знаходиться за адресою https://www.gnu.org/software/libc/manual/html_node/Opening-and-Closing-Files.html і вибрати 100 найбільш уживаних слів у тексті

38. Проаналізувати HTML текст із замітки газети NY Times (заув. HTML текст - це все що не є тегом), що знаходиться за адресою <http://www.cs.cornell.edu/courses/cs2043/2012sp/superbowl.html> (або іншу не малу сторінку) і дістати N слів, які з'являються або до або після будь-яких з наступних слів в тексті: "giants", "patriots"

39. Звичайний текст, в якому значення розділені комою(csv-file) містить лог активності деякого користувача, доступний за адресою http://www.cs.cornell.edu/courses/cs2043/2012sp/activity_log.csv . Лог має 3 стовпчики: дату, назву активності та час. Є три типи активності: work, run, farmers market. Порахувати загальну кількість годин, витрачених на кожну активність під час періоду в логі.

40. Скачайте наступний архів за допомогою відповідної команди, і розархівуйте його теж використовуючи оболонку <http://www.cs.cornell.edu/courses/cs2043/2012sp/hw4.tar.gz> . В цьому архіві знаходяться два файли tweets.txt який містить список із 2000 твітів, розділених новим рядком, другий файл впринципі для наших цілей не є потрібним (це є баш-скрипт який кожен твіт розміщає в новому файлі). Завдання полягає у написанні сі-шелл скрипта який приймає список слів і створює каталог для кожного слова, який містить всі твіти, що містять в собі це слово. Вміст каталогу - декілька файлів з одним твітом в кожному файлі. Додатково, якщо перший аргумент скрипта є -all то на виході повинний бути єдиний каталог, який містить твіти, які містять >всі< передані аргументи у скрипт.

В результаті виклику цього скрипта з аргументом restaurant - повинен утворитися каталог restaurant який міститиме всі файли з твітами що містять слово 'restaurant'.

Якщо викликати скрипт з двома аргументами - restaurant і ithaca повинно утворитися 2 каталоги з файлами, які містять твіти що відповідають відповідним ключовим словам.

В результаті виклику скрипта з аргументами -all restaurant ithaca повинен утворитися один каталог restaurant_ithaca який міститиме твіти, які містять обидва ключові слова одночасно.