

## Pliki – zadania:

1. Napisz program kopiujący zawartość pliku o nazwie podanej przez użytkownika jako pierwszy argument do pliku którego nazwa została podana jako drugi argument
2. Napisz program określający rozmiar plików których nazwy zostały wskazane przez użytkownika jako argumenty
3. Napisz program odwracający kolejność znaków w każdej linii pliku
4. Napisz program wyszukujący najdłuższą linię w pliku i podający ilość znaków w tej linii.
5. Napisz program wypisujący od końca zawartość pliku
6. Napisz procedurę kopiowania ostatnich 10 znaków, słów i ostatnich 10 linii pliku wskazanego pierwszym argumentem do nowo stworzonego pliku którego nazwa została wskazana drugim argumentem
7. Napisz program do rozpoznawania czy plik o podanej nazwie jest plikiem tekstowym (plik tekstowy zawiera znaki o kodach 0-127 – można w tym celu użyć funkcji `isascii`)
8. Napisz program konwertujący małe litery na duże w pliku o nazwie wskazanej jako argument. Jeśli nazwa pliku nie została podana, dane powinny zostać odczytane ze standardowego wejścia.
9. Napisz program sprawdzający czy zawartość dwóch plików wskazanych argumentami jest identyczna
10. Napisz program, który w pliku o nazwie podanej jako ostatni argument zapisze połączoną zawartość wszystkich plików których nazwy zostały podane w linii poleceń przed ostatnim argumentem.
11. Napisz program liczący ile jest słów w pliku (zakładamy, że słowa składają się z małych i dużych liter alfabetu oraz cyfr i znaku podkreślenia, a wszystkie pozostałe znaki są separatorami słów). Jeśli nazwa pliku nie została podana, dane powinny zostać odczytane ze standardowego wejścia.
12. Napisz program do wyznaczania częstości występowania liter w pliku tekstowym o nazwie podanej jako argument linii poleceń. Wynikiem działania programu powinien być wydruk na standardowym wyjściu określający procentową zawartość poszczególnych liter w całym tekście z pominięciem białych znaków oraz znaków interpunkcji.

13. Napisz program do filtrowania linii tekstu odczytywanego ze standardowego wejścia w taki sposób, że jeśli linia odczytanego tekstu zawiera łańcuch znaków przekazanych jako argument linii poleceń, to jest ona zapisywana na standardowym wyjściu, w przeciwnym przypadku jest pomijana.
14. Napisz program do znajdowania łańcucha znaków podanego jako pierwszy argument linii poleceń w plikach o nazwach podanych jako pozostałe argumenty linii poleceń. Program powinien informować o nazwie pliku i miejscu (numer linii, numer znaku w linii), w którym dany łańcuch został znaleziony.
15. Napisz program do formatowania akapitów tekstu w plikach o nazwach podanych jako argumenty linii poleceń w taki sposób, żeby długość linii (liczba znaków w linii) nie przekraczała wartości podanej jako pierwszy argument linii poleceń. Przyjmując, że w wyniku formatowania nie może nastąpić podział słowa pomiędzy dwa wiersze i że separatorem formatowanego akapitu jest pusta linia.
16. Napisz program do porównywania plików o nazwach przekazanych jako argumenty. Wynikiem działania programu ma być komunikat że *pliki są identyczne*, *pliki różnią się od znaku nr<nr znaku> w linii<nr znaku linii>* lub – gdy jeden z plików zawiera treść drugiego uzupełnioną o jakieś dodatkowe znaki – *plik <nazwa> zawiera<liczba>znaków więcej niż zawartość pliku <nazwa>*
17. Napisz program wyświetlający zawartość katalogu, którego nazwa została podana jako argument linii poleceń (można skorzystać z funkcji `getcwd` i `scandir`)
18. Napisz program wyświetlający informację o pliku, którego nazwa została podana jako argument linii poleceń (odpowiadający poleceniu `ls - l` (można skorzystać z funkcji `fstat`, `getpwuid`, `getgrgid`, `ctime`)).