

[Powrót \(Systemy operacyjne\)](#)

Programowanie systemowe - standardowe strumienie

ćwiczenia laboratoryjne

Przydatne typy

- [FILE](#) - wskaźnik na plik.

Przydatne biblioteki i funkcje

- [stdio.h](#)
 - [printf](#) - wypisuje tekst na ekran.
 - [perror](#) - wypisuje tekst na ekran (na strumień błędów).
 - [scanf](#) - pobiera dane od użytkownika.
 - [fopen](#) - otwiera plik.
 - [freopen](#) - otwiera plik ze wskazanym strumieniem.
 - [fprintf](#) - wpisuje tekst do pliku.
 - [fscanf](#) - pobiera wartość zmiennej z pliku.
 - [fgetc](#) - pobiera znak (na jednym bajcie) z pliku.
 - [fclose](#) - zamyka plik.
 - [popen](#) - rozpoczyna uruchomienie polecenia systemowe w tle i przekierowuje wyjście do strumienia potokowego.
 - [pclose](#) - zamyka strumień potokowy.
- [string.h](#)
 - [strcmp](#) - porównuje łańcuchy znaków i zwraca:
 - równą 0 jeżeli są takie same.
 - mniejszą niż 0 jeżeli pierwszy napis, poprzedza leksykograficznie drugi napis.
 - większą niż 0 jeżeli drugi napis, poprzedza leksykograficznie pierwszy napis.
 - [strcpy](#) - kopiuje łańcuch znaków (kopiuje również znak `\0`).
 - [strlen](#) - zwraca długość łańcucha (liczy liczbę znaków do momentu wystąpienia znaku `\0`).
- [stdlib.h](#)
 - [system](#) - wykonuje polecenie systemowe.
 - [malloc](#) - rezerwuje pamięć.
 - [calloc](#) - rezerwuje pamięć zerując bity.
 - [realloc](#) - zmienia rozmiar zarezerwowanej pamięci.
 - [free](#) - zwalnia obszar pamięci.

Standardowych strumienie

- [stdin](#) - standardowy strumień wejścia.
- [stdout](#) - standardowy strumień wyjścia.
- [stderr](#) - standardowy strumień wyjścia błędów.

Wartości specjalne

- [NULL](#) - znak wartości zerowej (możemy użyć w celu sprawdzenia czy plik udało się otworzyć, np. czy użytkownik nie pobrał złej ścieżki).
- [EOF](#) - znak końca pliku (możemy użyć w celu sprawdzenia czy pobrany znak nie jest końcem pliku).

Zadania

Zadanie 1. Napisz program, który odczyta dane z pliku [liczby.txt](#) oraz wyświetli największą i najmniejszą z nich na ekranie.

W przypadku problemów z otwarciem pliku, powinien pojawić się komunikat błędnie przekazany na strumieniu błędów.

Plik *liczby.txt* zawiera 100 liczb całkowitych.

Zadanie 2. Napisz program, który pobierze od użytkownika liczbę n oraz n liczb. Następnie wpisze do pliku *ujemne.txt* podane liczby ujemne oraz do pliku *dodatnie.txt* podane liczby dodatnie.

Jeżeli pojawi się liczba równa 0, powinien zostać wyświetlony odpowiedni komunikat na strumieniu błędów.

Zadanie 3. Napisz program, który pobierze od użytkownika dwa napisy. Następnie wpisze do pliku *przedzej.txt* napis, który jest prędzej

leksykograficznie oraz do pliku *dluzszy.txt* napis, który jest dłuższy.

Jeżeli dwa napisy są takiej samej długości powinien zostać wypisany komunikat o błędzie na strumieniu błędów.

Zadanie 4. Napisz program, który pobierze od użytkownika dowolne imię i nazwisko. Następnie sprawdzi czy w pliku [baza.txt](#) znajdują się osoba o podanych wartościach. Na ekranie powinien zostać wypisany odpowiedni komunikat. Jeżeli podane imię lub nazwisko zawiera inne znaki niż litery powinien zostać wyświetlony błąd na strumieniu błędów.

Plik *baza.txt* zawiera 70 wierszy.

Zadanie 5. Napisz program, który utworzy plik *przedmioty.txt* a następnie wpisze do niego zawartość:

informatyka

matematyka

jezyk angielski

Zadanie 6. Napisz program, który umożliwi:

- usunięcie
- skopiowanie do folderu *inne*
- przeniesienie do folderu *inne*

plik o nazwie *przedmioty.txt*.

Wykorzystaj polecenia systemowe.

Zadanie 7. Napisz program, który utworzy archiwum **.tar* lub **.zip* (w zależności od wyboru użytkownika) dla folderu o nazwie *temp*.

Wykorzystaj polecenia systemowe.

Zadanie 8. Napisz program, który wyświetli nazwy plików/folderów, które znajdują się w Twoim katalogu domowym oraz zawierają najdłuższą i najkrótszą nazwę.

Wykorzystaj strumień potokowy.

Zadanie 9. Napisz program, który wyświetli który spośród dwóch plików (nazwy dwóch plików dobierz wedle własnego uznania), ma więcej słów.

Wykorzystaj strumień potokowy.