

Wstęp do programowania. Lista 7. Pliki.

1. Napisz program, który wczyta i wyświetli tekst z pliku *cytaty.txt*.
2. Napisz program, który wyświetli co drugą linię z pliku *cytaty.txt*.
3. Napisz program, który wyświetli wybrany przez użytkownika zakres linii z pliku *cytaty.txt*. (np. od 2 do 13)
4. Napisz program, który zapisze w pliku przynajmniej dwie linie i je wyświetli.
5. Napisz program, który dopisze do pliku *nieskonczonosc.txt* przynajmniej dwie linie i wyświetli cały plik.
6. Napisz program, który wczyta tekst z pliku *nieskonczonosc.txt* i wyświetli go wielkimi literami.
7. Napisz program, który zapisze treść pliku *nieskonczonosc.txt* do nowego pliku.
8. Napisz program, który znajdzie i wyświetli najdłuższe słowo w pliku *nieskonczonosc.txt*. Następnie zmodyfikuj program, tak aby wyświetlał wszystkie wyrazy długości 5.
9. Napisz program, który w wyniku wyświetli ilość linii w pliku *cytaty.txt*
10. Napisz program, który w pliku *cytaty.txt* wstawi zamiast nowych linii znak tabulacji.
11. W pliku *zmienne.txt* podano w każdej linii kolejno wartości x, y, z . Stwórz nowy plik, który będzie zawierał w każdej linii wartości y, z, x .
12. Napisz program, który wyświetli nazwy plików z rozszerzeniem **.txt* znajdujące się w tym samym katalogu posortowane po dacie ostatniej modyfikacji. (zob. bibliotekę *os* i *path.getmtime* oraz bibliotekę *glob*)
13. Napisz program, który poda rozmiar pliku *cytaty.txt*.
14. Napisz program, wyszukujące hasła do krzyżówki. Użytkownik ma podać długość hasła oraz podać jakie litery w hasle zna. Program ma wypisać wszystkie pasujące hasła z pliku *word_list.txt*.
15. Napisz grę Państwo–stolica. Program ma najpierw wczytać imiona graczy. Następnie losowo ustalić kolejność. Program ma mieć 5 rund. Za każdym razem ma przypominać kto odpowiada na pytanie. Program ma pytać gracza o stolice jakiegoś państwa i sprawdzać poprawność odpowiedzi (uznawać nawet jeśli jest napisana małymi/wielkimi literami). Na koniec ma podać wyniki każdego gracza oraz wpisać je do pliku *wyniki.txt*. Na koniec program powinien podawać najlepszy uzyskany wyniki w całej historii istnienia programu.