

# Teoria informacji i kodowania - ćwiczenia laboratoryjne

## Zadanie 1

Napisać program w języku C++ wyznaczający model źródła informacji dla dowolnego pliku.

W tym celu należy policzyć korzystając z odpowiednio utworzonej tablicy ile razy w zadanym pliku wejściowym występują poszczególne symbole (bajty 8-bitowe) i zapisać w pliku wynikowym tabelę symboli wraz z odpowiadającą im liczbą wystąpień, posortowaną malejąco według liczby wystąpień.

### Szczegółowe wymagania stałe:

- program ma się poprawnie kompilować w środowisku Dev-C++ w wersji dostępnej w sali laboratoryjnej, w której prowadzone są zajęcia;
- cały program ma mieścić się w jednej jednostce kompilacyjnej (jednym pliku \*.cpp);
- program ma być uruchamiany z okna konsoli tekstowej, bez interfejsu graficznego;
- program ma rozpocząć i zakończyć swoje działanie bez potrzeby dodatkowych działań ze strony użytkownika (poza jego uruchomieniem); w szczególności bez oczekiwania na wciśnięcie dowolnego klawisza przez użytkownika;
- program ma informować na bieżąco użytkownika o wykonywanych operacjach w postaci informacji tekstowych wyświetlanych w oknie konsoli tekstowej;
- do obsługi wejścia/wyjścia i obsługi plików używać tylko strumieni, a do wszelkich łańcuchów znaków (tekstu) używać tylko typu string.

### Szczegółowe wymagania dotyczące zadania:

- nazwa pliku wejściowego ma być podawana jako parametr wejściowy przed uruchomieniem programu, plik wejściowy ma znajdować się w bieżącym katalogu roboczym;
- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem \*.ile ma być zapisana całkowita liczba symboli (bajtów 8-bitowych) występujących w pliku wejściowym;
- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem \*.licz ma być zapisana tabela symboli (bajtów 8-bitowych) wraz z liczbą wystąpień każdego symbolu w formacie:

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

...

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

Model źródła informacji dla danego pliku obejmuje tylko te symbole (bajty 8-bitowe), które występują w tym pliku czyli tylko te, których liczba wystąpień wynosi co najmniej 1.

- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem \*.msort ma być zapisana posortowana **malejąco** tabela symboli (bajtów 8-bitowych) wraz z liczbą wystąpień każdego symbolu w formacie:

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

...

*symbol (liczba naturalna w systemie dziesiętnym) <spacja> liczba wystąpień*

W przypadku, gdy dwa lub więcej symboli (bajtów 8-bitowych) mają taką samą liczbę wystąpień o kolejności w tabeli ma decydować wartość liczbową symbolu w porządku **narastającym**.

- do sortowania zastosować funkcje **qsort()** lub **sort()** albo samodzielnie zaimplementowane sortowanie bąbelkowe lub inne;
- w programie poza funkcją **main()** mają być także oddzielne funkcje z przekazywanymi do nich odpowiednimi argumentami co najmniej dla operacji liczenia liczby wystąpień i dla operacji sortowania tablicy symboli (bajtów 8-bitowych).

W wersji **podstawowej** jako odpowiednio utworzoną tablicę symboli (bajtów 8-bitowych) należy odpowiednio zastosować typ danych tablica statyczna.

W wersji **zaawansowanej** jako odpowiednio utworzoną tablicę symboli (bajtów 8-bitowych) należy zastosować listę dynamiczną zaimplementowaną samodzielnie lub skorzystać z gotowych kontenerów z biblioteki STL.