

# Teoria informacji i kodowania - ćwiczenia laboratoryjne

## Zadanie CRC

Napisać program w języku C++ wyliczający 32-bitowy cykliczny kod nadmiarowy CRC-32 dla dowolnego pliku i zapisać uzyskany wynik w pliku wynikowym.

W tym celu należy wyznaczyć zgodnie z algorytmem wyznaczania kodu CRC resztę z dzielenia ciągu danych (bajtów 8-bitowych) odczytywanego z zadanego pliku wejściowego przez 33-bitowy dzielnik CRC czyli wielomian generujący CRC. Należy użyć następującego wielomianu generującego zapisanego w standardowej postaci szesnastkowej: **0x04C11DB7**.

### 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 **\*.crc32** ma być zapisana wyznaczona wartość CRC-32 dla pliku wejściowego w formacie:

*CRC-32 w postaci tekstowej jako liczba w systemie dziesiętnym*

*CRC-32 w postaci tekstowej jako liczba w systemie szesnastkowym (8 znaków)*

*CRC-32 w postaci tekstowej jako liczba w systemie dwójkowym (32 znaki „0” lub „1”)*

*CRC-32 w postaci binarnej (4 bajty 8-bitowe)*

- w celu wykonania dzielenia przez wielomian należy wykorzystać operacje bitowe: negacja bitowa czyli NOT (operator ~), koniunkcja bitowa czyli AND (operator &), alternatywa bitowa czyli OR (operator |), dodawanie modulo 2 czyli XOR (operator ^), przesunięcie bitowe w lewo (operator <<), przesunięcie bitowe w prawo (operator >>).

Przy przesunięciach bitowych ważne jest sprawdzenie i upewnienie się jakie bity (0 czy 1) wstawiane są na wolne pozycje z prawej strony przy przesunięciu bitowym w lewo lub z lewej strony przy przesunięciu bitowym w prawo.

- bufor do wczytywania kolejnych danych z zadanego pliku wejściowego (z prawej strony) i wykonywania na nim dzielenia ma mieć rozmiar 32-bitowy i na początku ma zawierać **same bity 0**;

- po wczytaniu do bufora ostatniego bajtu danych mają być do niego jeszcze kolejno wpisane (z prawej strony) **32 bity 0**.
- w programie poza funkcją **main()** mają być także oddzielne funkcje z przekazywanymi do nich odpowiednimi parametrami co najmniej dla operacji wyliczania CRC-32 i zapisywania CRC-32 do pliku wynikowego w odpowiednich formatach.

W wersji **podstawowej** do wyznaczenia CRC-32 należy wykorzystać podstawowy algorytm dzielenia ciągu danych (bajtów 8-bitowych) odczytywanych z zadanego pliku wejściowego przez wielomian generujący CRC bit po bicie.

W wersji **zaawansowanej** do wyznaczenia CRC-32 należy wykorzystać algorytm tablicowy dzielenia ciągu danych (bajtów 8-bitowych) odczytywanych z zadanego pliku wejściowego przez wielomian generujący CRC umożliwiający dzielenie na raz całych bajtów 8-bitowych.