

Teoria informacji i kodowania - ćwiczenia laboratoryjne

Praca na zaliczenie

Na podstawie uprzednio wykonanych zadań napisać program w języku C++ dokonujący kompresji bezstratnej i dekompresji poprzez zastosowanie kodowania metodą Huffmana dla dowolnego pliku.

W tym celu należy odpowiednio połączyć w jeden program funkcje: wyznaczania modelu źródła informacji, wyznaczania drzewa kodowania i tabeli kodowej oraz kompresji i dekompresji, a także wyliczania 32-bitowego cyklicznego kodu nadmiarowego CRC-32.

Wymagania ogólne:

- 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;
- poszczególne etapy programu mają być wykonywane poza funkcją **main()** jako oddzielne funkcje z przekazywanymi do nich odpowiednimi argumentami.

Format uruchomienia programu:

<nazwa programu><spacja><tryb pracy><spacja><nazwa pliku>

<nazwa programu> - przyjęta nazwa własna programu do kompresji i dekompresji

<tryb pracy> - tryb pracy przyjmujący wartości: **/K** lub **/D**

/K – program będzie działał w trybie kompresji

/D – program będzie działał w trybie dekompresji

<nazwa pliku> - nazwa pliku wejściowego podawana jako parametr wejściowy, plik wejściowy ma znajdować się w bieżącym katalogu roboczym

dla trybu pracy **/K** – będzie to nazwa dowolnego pliku do kompresji

dla trybu pracy **/D** – będzie to nazwa pliku z rozszerzeniem ***.huff** do dekompresji

Przykładowe wywołania:

W celu wykonania kompresji: **huffcodec /K test1.txt**

W celu wykonania dekompresji: **huffcodec /D test1.huff**

Wynik działania programu w trybie kompresji

W wyniku działania programu w trybie kompresji w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego do kompresji lub innej dowolnej, ale w każdym przypadku z rozszerzeniem *.huff mają być zapisane dane po kompresji w formacie:

kod CRC-32	rozmiar tabeli kodowej	tabela kodowa	blok zakodowanych danych	liczba bitów znaczących ostatniego bajtu
-------------------	-------------------------------	----------------------	---------------------------------	---

kod CRC-32 – wyliczony dla pliku wejściowego 32-bitowy cykliczny kod nadmiarowy zapisany binarnie (4 bajty)

rozmiar tabeli kodowej – liczba pozycji tabeli kodowej wyznaczonej dla pliku wejściowego, inaczej liczba symboli alfabetu wejściowego, ale wyrażona indeksem ostatniej pozycji (wiersza) tabeli kodowej czyli jest to liczba symboli odjąć 1 (tak, aby zawsze otrzymać liczbę od 0 do 255) zapisana binarnie (1 bajt)

tabela kodowa – poszczególne pozycje (wiersze) tabeli kodowej wyznaczonej dla pliku wejściowego w liczbie określonej **rozmiarem tabeli kodowej** dodać 1; każda pozycja w formacie:

liczba bitów sekwencji kodującej	symbol	sekwencja kodująca
---	---------------	---------------------------

liczba bitów sekwencji kodującej – liczba bitów (pierwotnie liczba znaków w łańcuchu znaków kodujących (typu string) składających się z „0” i „1”) danej sekwencji kodującej zapisana binarnie (1 bajt)

symbol – dany symbol alfabetu wejściowego czyli liczba naturalna w systemie dziesiętnym zapisana binarnie (1 bajt)

sekwencja kodująca – sekwencja kodująca dla danego symbolu alfabetu wejściowego (pierwotnie wyrażona łańcuchem znaków kodujących (typu string) składających się z „0” i „1”) po uzupełnieniu zerami z prawej strony do sekwencji o długości będącej wielokrotnością 8 (rozmiar dla jednej pozycji w bajtach to liczba bitów uzupełniona do wielokrotności 8 podzielona przez 8)

np. dla sekwencji znakowej „1111 0100 1” będzie to bitowo 1111 0100 1000 0000 i zajmie 2 bajty;
do zamiany łańcuchów znaków kodujących z „0” i „1” na ich reprezentację binarną służy funkcja **strtol()**

blok zakodowanych danych – kolejno zakodowane zgodnie z tabelą kodową dane z pliku wejściowego

liczba bitów znaczących ostatniego bajtu – po zapisaniu wszystkich zakodowanych danych w pliku wynikowym na samym końcu tego pliku ma zostać zapisana informacja ile bitów jest ważnych w ostatnim bajcie bloku zakodowanych danych.

Wynik działania programu w trybie dekompresji

W wyniku działania programu w trybie dekompresji w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego do dekompresji lub innej dowolnej, ale w każdym przypadku z rozszerzeniem *.dhuff mają być zapisane dane po dekompresji.

Szczegółowy przebieg działania programu w trybie kompresji:

1. Sprawdzenie poprawności parametrów podanych przed uruchomieniem programu
 - a) w przypadku poprawnych parametrów wyświetlenie informacji o trybie pracy programu;
 - b) w przypadku błędnych parametrów wyświetlenie komunikatu z informacją o poprawnym formacie uruchomienia programu i zakończenie wykonywania programu;
2. Wyświetlenie informacji z nazwą pliku do kompresji
3. Wyliczenie kodu CRC-32 dla pliku wejściowego
 - a) wyświetlenie informacji o rozpoczęciu wyliczania kodu CRC-32;
 - b) wyliczenie kodu CRC-32 i zapisanie go do pliku wynikowego;
 - c) wyświetlenie informacji z kodem CRC-32, który został wyliczony;
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
4. Tworzenie modelu źródła informacji dla pliku wejściowego
 - a) wyświetlenie informacji o rozpoczęciu tworzenia modelu źródła informacji;
 - b) utworzenie modelu źródła informacji;
 - c) wyświetlenie informacji z wyznaczoną liczbą symboli alfabetu wejściowego;
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
5. Tworzenie drzewa kodowania dla wyznaczonego modelu źródła informacji
 - a) wyświetlenie informacji o rozpoczęciu tworzenia drzewa kodowania;
 - b) utworzenie drzewa kodowania;
 - c) wyświetlenie informacji z wyznaczoną liczbą symboli zastępczych;
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
6. Tworzenie tabeli kodowej dla wyznaczonego drzewa kodowania
 - a) wyświetlenie informacji o rozpoczęciu tworzenia tabeli kodowej;
 - b) utworzenie tabeli kodowej i odpowiednie zapisanie jej do pliku wynikowego wraz z rozmiarem (wyrażonym indeksem ostatniej pozycji);
 - c) wyświetlenie informacji z rozmiarem tabeli kodowej (indeks ostatniej pozycji dodać 1);
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
7. Kompresja danych z pliku wejściowego na podstawie wyznaczonej tabeli kodowej
 - a) wyświetlenie informacji o rozpoczęciu kompresji;
 - b) tworzenie bloku zakodowanych danych i sukcesywne zapisywanie go do pliku wynikowego;
 - c) po zapisaniu wszystkich zakodowanych danych w pliku wynikowym na samym jego końcu zapisanie informacji ile bitów jest ważnych w ostatnim bajcie bloku zakodowanych danych;
 - d) wyświetlenie informacji z liczbą bajtów bloku zakodowanych danych (tylko samego bloku);
 - e) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
8. Wyświetlenie informacji z nazwą pliku po kompresji i wynikami kompresji
 - a) wyświetlenie informacji z całkowitym rozmiarem pliku skompresowanego i pliku wejściowego;
 - b) wyświetlenie informacji ze współczynnikiem kompresji wyznaczonym jako iloraz rozmiaru pliku skompresowanego przez rozmiar pliku wejściowego wyrażony procentowo;
9. Informacja o poprawnym zakończeniu programu w trybie kompresji.

Szczegółowy przebieg działania programu w trybie dekompresji:

1. Sprawdzenie poprawności parametrów podanych przed uruchomieniem programu
 - a) w przypadku poprawnych parametrów wyświetlenie informacji o trybie pracy programu;
 - b) w przypadku błędnych parametrów wyświetlenie komunikatu z informacją o poprawnym formacie uruchomienia programu i zakończenie wykonywania programu;
2. Wyświetlenie informacji z nazwą pliku do dekompresji
3. Odczytanie kodu CRC-32 z pliku wejściowego
 - a) wyświetlenie informacji o rozpoczęciu odczytywania kodu CRC-32;
 - b) odczytanie kodu CRC-32 z pliku wejściowego;
 - c) wyświetlenie informacji z kodem CRC-32, który został odczytany;
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
4. Odczytanie tabeli kodowej z pliku wejściowego
 - a) wyświetlenie informacji o rozpoczęciu odczytywania tabeli kodowej;
 - b) odczytanie tabeli kodowej z pliku wejściowego wraz z jej rozmiarem;
 - c) wyświetlenie informacji z liczbą symboli alfabetu wejściowego (wyrażonym indeksem ostatniej pozycji tabeli kodowej dodać 1);
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
5. Dekompresja danych z pliku wejściowego na podstawie odczytanej tabeli kodowej
 - a) wyświetlenie informacji o rozpoczęciu dekompresji;
 - b) dekodowanie danych i zapisywanie ich do pliku wynikowego;
 - c) wyświetlenie informacji z liczbą odkodowanych bajtów (rozmiarem pliku wynikowego);
 - d) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
6. Wyświetlenie informacji z nazwą pliku po dekompresji
7. Wyliczenie kodu CRC-32 dla pliku wynikowego
 - a) wyświetlenie informacji o rozpoczęciu wyliczania kodu CRC-32;
 - b) wyliczenie kodu CRC-32;
 - c) wyświetlenie informacji z kodem CRC-32, który został wyliczony;
 - d) porównanie wyliczonego kodu CRC-32 dla pliku wynikowego z kodem CRC-32 odczytanym z pliku wejściowego i wyświetlenie informacji z wynikiem tego porównania;
 - e) w przypadku błędu wyświetlenie komunikatu o błędzie i zakończenie programu;
8. Informacja o poprawnym zakończeniu programu w trybie dekompresji.