

Rafał Szymanek  
Nr indeksu: 241436  
14 Stycznia 2018

# Dokumentacja

## Program kompresujący tekst algorytmem RLE

### Dokumentacja użytkownika

Program ten jest aplikacją terminalową. Obsługujemy ją klawiaturą wykonując polecenia wskazane przez program. Działa on zarówno na komputerach z systemem Microsoft Windows, a także na systemach opierających się o jądro UNIX (Linux, Mac OS).

**Uwaga wstępna:** Na systemie Windows przy wyborze menu nie musimy wcisnąć klawisza enter poruszając się po Menu Głównym.

Pierwsze co pokazuje nam się na ekranie jest Menu Główne wyboru:

- 1) Skompresuj gotowy plik
- 2) Skompresuj wpisaną frazę z klawiatury
- 3) Dekompresja pliku
- 4) Dekompresja wpisanej frazy z klawiatury
- 5) Wyjdź z programu

### Skompresuj gotowy plik

Wybierając tą opcję musimy mieć gotowy plik “.txt”, a także znać jego ścieżkę. Po wybraniu tej opcji program zapyta nas o ilość linijek tekstu w naszym Istotnym krokiem jest podanie poprawnej ścieżki do pliku.

Mac/Linux: /Users/Rafał\_Szymanek/Documents/PlikiProjekt/plik.txt

Windows: C:\katalog1\katalog2\plik.txt

Jeżeli wszystko będzie poprawnie działało, program przepuści nas dalej, jeżeli zaś wystąpią błędy, wrócimy z powrotem do podania ścieżki do pliku.

Ukaże nam się także poziom kompresji naszego pliku.

Następnym krokiem będzie wybór co chcemy zrobić z danym skompresowanym tekstem:

- 1) Nadpisz plik
- 2) Zapisz w nowym pliku
- 3) Tylko wyświetl skompresowane dane

**Nadpisz plik** - Nadpisujemy plik, który podaliśmy na początku programu. Utracimy oryginalny plik! Po zapisaniu wrócimy do menu głównego.

**Zapisz w nowym pliku** - Podajemy nową ścieżkę, z nową nazwą na samym końcu.

Uwaga! Jeżeli podamy taką samą ścieżkę jak na początku programu z tą samą nazwą pliku, również utracimy oryginalny plik! Po zapisaniu wrócimy do menu głównego.

**Tylko wyświetl skompresowane dane** - Opcja ta pokazuje nam skompresowany tekst bez, żadnego zapisywania. Kiedy skończymy przeglądanie tekstu należy wcisnąć klawisz "Enter". W ten sposób powrócimy do Menu Głównego.

## Skompresuj wpisaną frazę z klawiatury

Wybierając tą opcję program zapyta nas o frazę jaką chcemy skompresować. Uwaga klawisz "Enter" kończy wpisywanie frazy tekstu.

Ukaże nam się także poziom kompresji naszego pliku.

Pojawi nam się kolejne menu

- 1) Zapisz w nowym pliku
- 2) Tylko wyświetl skompresowane dane

**Zapisz w nowym pliku** - Program zapyta nas o ścieżkę do zapisania danej frazy. Następnie przeniesie nas do Menu Głównego.

Tylko wyświetl skompresowane dane - Program wyświetla nam skompresowaną frazę, a podgląd kończymy klawiszem "Enter", po czym wracamy do Menu Głównego.

## Dekompresja pliku

Opcja ta dekompresuje nam plik wcześniej skompresowany algorytmem RLE. Ważne jest aby jako znaku specjalnego w kompresji użyć znaku ":".

Program zapyta nas o ścieżkę do pliku, w wypadku niepoprawnej ścieżki poprosi o nią ponownie.

Kiedy dane zostaną zdekompresowane, mamy do wyboru:

- 1) Nadpisz plik
- 2) Zapisz w nowym pliku
- 3) Tylko wyświetl skompresowane dane

**Nadpisz plik** - nadpisuje ścieżkę, którą podaliśmy przy jako plik źródłowy. Tracimy oryginalny plik!

**Zapisz w nowym pliku** - Program pyta nas o ścieżkę, jeżeli podamy ją błędnie zapyta ponownie. Zapisuje w nowym pliku. Uwaga, jeżeli podamy tą samą ścieżkę jak plik źródłowy utracimy oryginalny plik!

**Tylko wyświetl skompresowane dane** - wyświetla nam dane, których nie chcemy zapisać.

## Dekompresja wpisanej frazy z klawiatury

Program dekompresuje nam wpisaną frazę z klawiatury, zapisaną algorytmem RLE, gdzie separatorem jest “:”. Na początku zostaniemy poproszeni o wpisanie frazy. Po tym dane zostaną dekompresowane i zostaniemy zapytani co chcemy z nimi zrobić:

- 1) Zapisz w nowym pliku
- 2) Tylko wyświetl zdekompresowane dane

**Zapisz w nowym pliku** - Zapisujemy rezultat operacji do nowego pliku, program zapyta nas o ścieżkę, gdzie mamy zapisać dany plik. Po zakończonym działaniu ujrzymy Menu Główne.

**Tylko wyświetl zdekompresowane dane** - Wyświetla nam wynik dekompresji, kiedy zakończymy przeglądanie należy kliknąć “Enter”, przeniesie nas to Menu Głównego.

## Wyjdź z programu

Po wybraniu tej opcji po trzech sekundach nastąpi wyłączenie programu.

# Dokumentacja techniczna

Techniczny opis działania programu i wytłumaczenie działania poszczególnych funkcji.

Załączone biblioteki:

**iostream** - Podstawowa biblioteka przyjmowania danych i wypisywania ich na ekran.

**fstream** - Biblioteka, która umożliwia zapis i odczyt z pliku.

**math.h** - Biblioteka matematyczna, służy mi do funkcji floor();

**string** - Służy do operacji na zmiennych typu string.

**cstdlib** - Dla Windows, aby można było użyć komand systemowych ("cls") oraz funkcji sleep.

**conio.h** - Umożliwia (tylko na Windowsie), opcje, że nie trzeba wciskać klawisza "Enter" po wyborze opcji w menu.

**windows.h** - Umożliwia wpisywanie komend do konsoli systemu Windows.

**sstream** - Służy do konwersji liczby typu int na zmienną typu string.

**unistd.h** - Biblioteka dla UNIX-a, dla funkcji m.in sleep();

**stdlib.h** - Biblioteka dla UNIX-a, do wydawania poleceń systemowych np. ("clear")

```
#ifdef _WIN32
```

```
    // polecenia wykonywane tylko na systemie Windows
```

```
#else
```

```
    // polecenia wykonywane na innych systemach niż Windows
```

```
#endif
```

## Funkcje

**NumberToString ( T Number )** - używamy do zamiany zmiennej typu int na zmienną typu string. Funkcja jest potrzebna tylko na systemie Windows. Na Unix służy do tego funkcja to\_string();.

**void clearScreen()** - jest to funkcja do czyszczenia ekranu w konsoli. Zrobiona jest z tego powodu, aby działało to na obu systemach.

**void sleepForSeconds(int n)** - jest to funkcja usypiania ekranu na określoną "n" liczbę czasu. Parametr "n" jest właśnie liczbą sekund w Unix-ie, a w Windowsie milisekund, na ile program ma zostać uśpiony.

**char getCharacter()** - Pobierz jeden znak. Funkcja jest stworzona po to aby w Windowsie nie musieć klikać "Enter" po wyborze numeru z menu. Nie ma niestety odpowiednika na Unix-ie. Zwraca nam to co wpisał użytkownik.

### **Uwaga do funkcji, których zadaniem jest otwieranie i zapisywanie do pliku :**

Różni się tutaj sposób wpisania ścieżki. Na systemie Unix możemy użyć do tego zwykłego stringa. Na Windowsie zaś musimy przekonwertować zmienną typu string funkcją `c_str()`;

**int LiczenieLiczbyLinijekPliku(string \*sciezka)** - Funkcja ta wyręcza użytkownika przed wpisywaniem ile linijek ma jego plik ".txt". Do działania programu jako parametr musimy podać ścieżkę do pliku, jaką wpisał użytkownik. Jako wynik zwraca właśnie ile linijek ma wczytany plik. Sprawdza czy plik się dobrze otwiera, a następnie za pomocą pętli `while()` i zmiennej pomocniczej "licznik", sprawdzamy ile linijek ma nasz plik.

**int WczytywanieDanychZPliku(string \*sciezka, string \*tab)** - Funkcja ta wczytuje dane z pliku i zapisuje je do tablicy(używamy wskaźnika). Jeżeli plik się nie otworzy zwrócone zostanie "1", jeżeli zaś otwarcie będzie pozytywne uzyskamy wynik "0".

**int ZapisDoPliku(string \*tab, int liczbaLinijek, string sciezka)** - Funkcja zapisuje wynik naszej operacji do pliku. Potrzebna nam jest tablica, którą chcemy zapisać, `liczbaLinijek` czyli tak naprawdę ile "szufladek" mamy w tablicy oraz ścieżka do samej tablicy. Jeżeli całość się powiedzie dostaniemy wynik "0", jeżeli zaś nie to dostaniemy "1".

**int ZapisDoPliku(string linia, string sciezka)** - Funkcja zapisuje wynik naszej operacji do pliku. Na wejściu podajemy linię, a także ścieżkę. Jeżeli całość się powiedzie dostaniemy wynik "0", jeżeli zaś nie to dostaniemy "1".

**string ZamianaCiagowZnakow(string linia)** - Funkcja ta jest częścią kompresji. Znajduje ona kolejne takie same znaki i zamienia je na postać litera:ilość np. a:3. Za argument przyjmuje daną frazę, na której mamy pracować. Zwraca nam skompresowany tekst.

**void Kompresja(string \*tab, int liczbaLinijek, string \*tabDocelowa)** - Zapisuje nam skompresowany tekst do nowej tablicy, gdzie znajdują się skompresowane dane. Argumentami są tablica skąd bierzemy dane, `liczbaLinijek` z pliku źródłowego, tablica gdzie mają znaleźć się skompresowane dane.

**bool is\_number(const std::string& s)** - Znalazłem gotową funkcję w internecie. Sprawdza ona czy dany string jest numerem. Pomocna funkcja przy dekompresji.

**string Dekompresja(string fraza)** - Funkcja, która nam dekompresuje całą frazę zapisaną algorytmem RLE. Podajemy frazę do skompresowania i na wyjściu dostajemy zdekompresowaną frazę.

**int sprawdzanieDlugosciZnakowWTablicy(string \*tab,int &liczbaLinijek)** - Sprawdza ile znaków mamy w danej tablicy. Jako argumenty bierze tablicę oraz liczbę linijek

z naszego pliku. Jako wynik zwraca ilość wszystkich znaków występujących w całej tablicy. Funkcja jest dodatkową funkcją do liczenia % kompresji pliku.

**float liczeniePoziomuKompresji(int przedKompresja, int poKompresji) -**

Funkcja ta liczy ile % udało nam się uzyskać na kompresji. Jako argumenty bierze liczbę znaków przed i po kompresji. Jako wynik zwraca float zaokrąglony do dwóch miejsc po przecinku.

**Funkcja main()**

Służy on do interakcji z użytkownikiem, wyświetla wszystkie menu, informuje nas o błędach, i dba o to aby użytkownik nie wprowadził błędnych danych.

Do zapętlenia menu służą tu pętle while(której argumentem jest zmienna cnd [ciąg dalszy]), oprócz menu głównego, które zapętłone jest nieskończona pętlą for(;;). Za menu służy instrukcja warunkowa switch. Za zapętlenie wpisywania ścieżki służy nieskończona pętla for(;;).

Stosowane są tutaj tablice dynamiczne, w celu dobrego zarządzania niepotrzebną już pamięcią.