

**Dokumentacja funkcjonalna**  
Kompresor plików z wykorzystaniem  
algorytmu Huffmana

**Łukasz Gumienniczuk, Paweł Rusak**

**11 kwietnia 2023**

## Spis treści

<b>1</b>	<b>Kompilacja programu . . . . .</b>	<b>2</b>
<b>2</b>	<b>Flagi . . . . .</b>	<b>2</b>
<b>3</b>	<b>Instrukcja użycia . . . . .</b>	<b>3</b>
<b>4</b>	<b>Schemat pliku skompresowanego . . . . .</b>	<b>4</b>

## 1 Kompilacja programu

Program należy skompilować w katalogu głównym projektu w środowisku Linux. Można zrobić to za pomocą komendy `make`, która kompiluje program.

Program można także jednocześnie skompilować i przetestować na różnych plikach i różnych trybach. Poniżej znajdują się lista opcji kompilacji.

- `make test1` – kompilacja programu z testem na pliku `.txt` zawierającym treść „Pana Tadeusza” na 8 bitach,
- `make test2` – kompilacja programu z testem na pliku `.txt` zawierającym treść „Pana Tadeusza” na 12 bitach,
- `make test3` – kompilacja programu z testem na pliku `.txt` zawierającym treść „Pana Tadeusza” na 16 bitach,
- `make test4` – kompilacja programu z testem na pliku `.jpeg` przedstawiającym zdjęcie wiewiórki na 8 bitach,
- `make test5` – kompilacja programu z testem na pliku `.jpeg` przedstawiającym zdjęcie wiewiórki na 12 bitach,
- `make test6` – kompilacja programu z testem na pliku `.jpeg` przedstawiającym zdjęcie wiewiórki na 16 bitach.

## 2 Flagi

Program obsługuje flagi zawarte w poniższej tabeli. Flagę lub flagi należy podać przy wywołaniu programu po pliku wejściowym i wyjściowym.

Przykładowe wywołanie programu wraz z flagami znajduje się w sekcji

Flaga	Opis flagi
-8	Program przeprowadzi kompresję na 8 bitach
-12	Program przeprowadzi kompresję na 12 bitach
-16	Program przeprowadzi kompresję na 16 bitach
-help	Program wyświetli instrukcje obsługi i zakończy działanie
-v	Program dodatkowo wyświetli słownik
-c	Program będzie działał w trybie kompresora
-x	Program będzie działał w trybie dekompresora

### 3 Instrukcja użycia

Poniżej prezentujemy sposób uruchomienia skompilowanego programu:

```
./program in_file out_file <flags>
```

Przykłady:

- `./program PanTadeusz.txt compress.txt --12 --v,`
- `./program compress.txt rezultat.txt --x,`
- `./program zdjecie.jpg skompresowany_jpg.txt --8 --v --c`

Uwaga:

- Jeżeli program będzie wywołany bez flag `--c` oraz `--v` to automatycznie rozpozna czy plik wejściowy jest plikiem skompresowanym czy nieskompresowanym.
- Flag `--8`, `--12` oraz `--16` należy używać tylko przy kompresji.
- Brak flagi wybierającej długość sekwencji bitowej przy kompresji będzie skutkowało kompresją na 8 bitach.
- Jeżeli plik jest jednocześnie plikiem wejściowym i wyjściowym to plik ten zostanie nadpisany wynikiem działania programu.
- Pojawienie się flagi `--help` skutkuje wyświetleniem pomocy użytkownika i zakończeniem działania programu.
- Pojawienie się nieznanej flagi skutkuje przerwaniem działania programu.

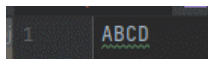
## 4 Schemat pliku skompresowanego



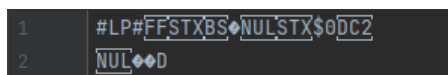
1. Nagłówek – #LP#,
2. Stopień kompresji zapisany binarne,
3. Reszta1 – liczba bitów, które czytamy w ostatnim lub przedostatnim bajcie (w zależności czy reszta2 jest różna od 0),
4. Reszta2 – liczba bitów, czytana w końcowym bajcie, jeżeli reszta2 jest różna od zera. Otrzymujemy ją podczas dzielenia danych na 12 lub 16 bitów (w zależności od stopnia kompresji). W 8 nie występuje. *Nie jest kompresowana.* ,
5. Suma kontrolna – jeżeli jej XOR przez DANE jest równy 0 to znaczy, że plik jest poprawny,
6. Liczba słów w słowniku – zapisana na dwóch bajtach (gdyby liczba przekraczała 256),
7. ZNAK[8/12/16] – bity, które kodowaliśmy,
8. Długość kodu – zajmuje cały 1 bajt,
9. Kod – *Jeżeli jest to ostatni kod, to dopełniamy zerami do pełnego bajta.*

Przykład:

- Plik kompresowany:



- Plik skompresowany:



Przykład pliku skompresowanego (zapis binarny):

