

ProgramowanieProceduralne



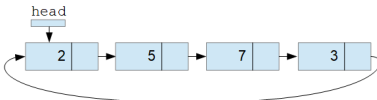
[Strona główna](#) / [Moje kursy](#) / [PP](#) / [LAB 14](#) / [IS L9](#)

IS_L9

1. Dana jest struktura

```
struct tnode {
    int value;
    struct tnode *next;
};
```

Lista jednokierunkowa cykliczna (ostatni element listy wskazuje na element pierwszy)

- pusta 
- jednoelementowa 
- wieloelementowa 

Proszę napisać program (złożony z trzech plików `cycle.h`, `cycle.c`, `cycle_main.c`) implementujący obsługę listy jednokierunkowej cyklicznej. Program powinien realizować następujące funkcjonalności - w żadnej z funkcji nie tworzymy nowych elementów :

- (1) wypisywanie listy - `void print_list(struct tnode*);`
- (1) dodawanie elementu do listy - `struct tnode* add_el(struct tnode* head, struct tnode* el);`
- (1) usuwanie elementu z listy (wczytanie wartości klucza do wyboru elementu), jeżeli mamy kilka elementów o podanym kluczu usuwamy wszystkie -
`void del_el(struct tnode** head, int var);`
- (1) łączenie dwóch list cyklicznych w jedną - `struct tnode* add_list(struct tnode* head1, struct tnode* head2);`
- (2) podział listy na dwie listy cykliczne - wartości parzyste i nieparzyste - `struct tnode* div_list(struct tnode** head);`
- (2) odwracanie listy w miejscu - `void rev_list(struct tnode*);`
- (1) zwalnianie listy - `void free_list(struct tnode**);`

(1) Proszę napisać plik `makefile` korzystający z metod jawnych (w odpowiednich miejscach proszę użyć dyrektywy `#ifndef`).

2. (3) Proszę napisać program, który tworzy skorowidz (plik wyjściowy) przy użyciu listy jednokierunkowej na podstawie pliku (plik wejściowy), którego nazwa podana jest jako **argument wywołania**.

Skorowidz powinien zawierać alfabetyczną listę wystąpień wszystkich słów w danym tekście, wraz z liczbą wystąpień danego słowa (przez słowo rozumiemy każdy ciąg znaków niezawierający znaków "białych" tzn. spacji, tabulatora, znaku nowej linii itd. czyli tak jak wczytuje je funkcja `scanf`).

Zatem dla tekstu: **qq abc qq ala ma kota abc abc**

program powinien wyświetlić:

```
abc: 3
ala: 1
kota: 1
ma: 1
qq: 2
```






Struktura w zadaniu powinna zawierać trzy pola:

- tablicę do przechowywania jednego słowa - trzeba założyć maksymalną długość słowa,
- licznik ile razy słowo przechowywane w tablicy wystąpiło we wczytywanym pliku,
- wskaźnik do następnej struktury.

Najlepiej tworzyć listę posortowaną względem przechowywanego słowa - ułatwienie w wyszukiwaniu czy słowo było już wczytane. Jeśli w liście wystąpiło już takie słowo (czyli było już wcześniej we wczytanym tekście), wówczas tylko zwiększamy licznik w odpowiadającej mu strukturze o 1. Jeżeli słowo pojawia się po raz pierwszy tworzymy dla niego nowy element listy, licznik ustawiamy na 1 i wstawiamy w odpowiednim miejscu listy.

Program proszę przetestować na pliku: [plik_wejscowy](#) i porównać wynik z rozwiązaniem [plik_wyjscowy](#).

Status przesłanego zadania

| | |
|---|--|
| Status przesłanego zadania | Przesłane do oceny |
| Stan oceniania | Nieocenione |
| Termin oddania | poniedziałek, 1 czerwca 2020, 14:25 |
| Pozostały czas | Zadanie zostało złożone 8 min. 18 sek. przed terminem |
| Ostatnio modyfikowane | poniedziałek, 1 czerwca 2020, 14:16 |
| Przesyłane pliki | <div><div><div>-</div><div></div><div>cycle_main.c</div><div>1 czerwca 2020, 14:16</div></div><div><div>-</div><div></div><div>cycle.c</div><div>1 czerwca 2020, 14:16</div></div><div><div>-</div><div></div><div>cycle.h</div><div>1 czerwca 2020, 14:16</div></div><div><div>-</div><div></div><div>makefile.txt</div><div>1 czerwca 2020, 14:16</div></div><div><div>-</div><div></div><div>zad2.c</div><div>1 czerwca 2020, 14:16</div></div></div> |
| Komentarz do przesłanego zadania | Komentarze (1) |

[◀ IS_L9](#)[LAB_15 ▶](#)

Platforma e-Learningowa obsługiwana jest przez:
Centrum e-Learningu AGH oraz Centrum Rozwiązań Informatycznych AGH

[Pobierz aplikację mobilną](#)