Rok akademicki 2019/20 **Informatyka - S1 - semestr 3**

Algorytmy 2

Laboratorium 2 – Lista jednokierunkowa (nie cykliczna)

Celem zadania jest implementacja słownika w postaci dynamicznej jednokierunkowej listy nie cyklicznej.
Implementacja nie może odwoływać się do typu tablicowego, lecz musi się posługiwać **wskaźnikami (referencjami do kolejnych węzłów listy)**.
Każdy z węzłów listy posiada następujące składowe:

- pole składowej kluczowej typu całkowitego (**int**);
- pole typu **double**;
- pole typu **char**;
- wskaźnik na węzeł będący następnikiem;

Składowa kluczowa węzła będącego pierwszym węzłem listy (o ile lista nie jest pusta) jest najmniejszą spośród składowych kluczowych wszystkich węzłów znajdujących się w liście, tzn. elementy listy są zawsze uporządkowane rosnąco ze względu na wartość składowej kluczowej;

Klucze (składowe kluczowe) są **unikalne** w ramach listy.
Należy stworzyć funkcję realizujące:

1) wstawienie do listy nowego elementu (uwzględnić przypadek, w którym węzeł o zadanej wartości składowej kluczowej już znajduje się w liście, należy wtedy zasygnalizować błąd); funkcja ustawia pole typu **double** na losowe wartości, zaś pole typu **char** na wartość **'T'**;
2) wstawienie do listy ***X*** nowych elementów o wygenerowanych losowo i różnych wartościach składowych kluczowych z zakresu 99 do 99999 (wartość ***X*** podana jako argument funkcji);
3) wyszukanie w liście elementu o podanej jako argument wartości składowej kluczowej (wraz z obsługą przypadku, w którym element nie istnieje);
4) usunięcie z listy elementu o podanej jako argument wartości składowej kluczowej (uwzględniające obsługę przypadku, w którym taki element nie istnieje);
5) prezentacja wartości składowych kluczowych pierwszych ***Y*** (wartość ***Y*** jest argumentem funkcji prezentacji) węzłów znajdujących się w liście (począwszy od składowej kluczowej węzła będącego czołem listy);
6) prezentacja wartości składowych kluczowych ostatnich ***Z*** (wartość ***Z*** jest argumentem funkcji prezentacji) węzłów znajdujących się w liście (począwszy od składowej kluczowej węzła będącego ogonem (ostatnim węzłem) listy);
7) wypisywanie liczby węzłów znajdujących się na liście (można utworzyć dodatkową zmienną do przechowywania liczby elementów listy);
8) usuwanie wszystkich elementów listy;

Listę można zaimplementować strukturalnie (C) lub obiektowo (C++);
Na liście argumentów każdej z wymienionych funkcji mogą się pojawić wyłącznie:

- referencja (wskaźnik) do pierwszego węzła (czoła) listy *(tylko w przypadku implementacji strukturalnej);*
- wartość składowej kluczowej - za wyjątkiem funkcji wstawiania *X* elementów, która zamiast wartości składowej kluczowej zawiera liczbę elementów do wylosowania (*w przypadku implementacji strukturalnej i obiektowej*) oraz funkcji prezentacji zawierających dodatkowo liczbę elementów do wyświetlenia;

Wszystkie funkcje powinny obsługiwać błędy (np. funkcja prezentacji powinna wypisywać odpowiedni komunikat, gdy lista jest pusta lub nie istnieje).

Program po uruchomieniu wczytuje plik wejściowy `inlab02.txt`**.**
Plik `inlab02.txt` zawiera w pierwszej linii kolejno liczbę elementów do wylosowania **X**, a następnie wartości pięciu kluczy **k1**, **k2**, **k3, k4, k5.**
Następnie wywoływana jest sekwencja funkcji (dalej w funkcji main()):

- czas start;
- zainicjuj listę;
- wyszukanie klucza **k1**;
- wstawienie **X** elementów do listy;
- wypisz liczbę węzłów w liście;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- wstaw element o wartości klucza **k2**;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- wstaw element o wartości klucza **k3**;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- wstaw element o wartości klucza **k4**;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- wstaw element o wartości klucza **k5**;
- usuń element o wartości klucza **k3**;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- usuń element o wartości klucza **k2**;
- prezentacja wartości kluczowych pierwszych 20 węzłów począwszy od czoła listy;
- usuń element o wartości klucza **k5**;
- wypisz liczbę węzłów w liście;
- wyszukaj element o wartości klucza **k5**;
- prezentacja wartości kluczowych ostatnich 11 węzłów;
- wypisz liczbę węzłów w liście;
- usuń wszystkie elementy listy;
- czas stop;
- wypisz czas wykonania.

Przygotowanie e-maila do wysłania:

Uwaga! Kod źródłowy programu (1 plik) po oddaniu prowadzącemu zajęcia laboratoryjne musi zostać przesłany na adres `algo2@zut.edu.pl` :

- plik z kodem źródłowym musi mieć nazwę: `nr_albumu.algo2.lab02.main.c` (np. `123456.algo2.lab02.main.c`); jeśli kod źródłowy programu składa się z wielu plików, to należy stworzyć jeden plik, umieszczając w nim kody wszystkich plików składowych; (plik może mieć rozszerzenie `.c` lub `.cpp`)
- plik musi zostać wysłany z poczty ZUT (zut.edu.pl);
- nagłówek maila (temat) musi mieć postać: `ALGO2 IS1 XXXY LAB02`, gdzie `XXXY` to numer grupy (np. `ALGO2 IS1 210C LAB02`) ;
- w pierwszych trzech liniach pliku z kodem źródłowym w komentarzach muszą znaleźć się:
  - linia 1: informacja identyczna z zamieszczoną w nagłówku maila
  - linia 2: imię i nazwisko
  - linia 3: adres email
- email **nie** powinien zawierać żadnej treści (tylko załącznik).