Rok akademicki 2017/18 **Informatyka - S1 - semestr 3**

Struktury danych i złożoność obliczeniowa

Laboratorium 1 – Tablica struktur

Zdefiniuj typ strukturalny (odpowiednik abstrakcyjnego typu rekordowego) zawierający 3 pola, odpowiednio**: *int, char, float***. Następnie zaimplementuj funkcje do obsługi zdefiniowanego typu:

- *losowanie* – funkcja:
  - pobiera jako argument liczbę **N** struktur, które mają zostać utworzone;
  - dynamicznie alokuje pamięć na tablicę **N** wskaźników na struktury;
  - następnie alokuje kolejno **N** struktur, przypisując uzyskane adresy do kolejnych komórek utworzonej wcześniej tablicy;
  - pole typu ***int*** jest ustawiane na wartość losową pomiędzy **-1000** a **9000**; pole typu ***char*** jest ustawiane na losową literę z zakresu od ***B*** do **X**; zaś pole typu ***float*** jest ustawiane na wartość **1000 + numer kolejny struktury** (od **1** do **N**);
  - funkcja zwraca adres tablicy.

  UWAGA: Zadbaj o to, by w każdym kolejnym uruchomieniu programu wartości losowe były odmienne (np. wykorzystując funkcję biblioteczną **srand()**), oraz o to, by każda spośród **N** struktur miała inną wartość pola typu ***int***.
- *kasowanie* – funkcja:
  - pobiera wskaźnik na tablicę struktur i jej wielkość (liczba przechowywanych wskaźników na struktury);
  - zwalniana jest najpierw kolejno pamięć zajęta przez wszystkie przechowywane struktury;
  - następnie zwalniana jest również pamięć zajęta przez samą tablicę wskaźników na struktury.
- *sortowanie* – funkcja:
  - pobiera wskaźnik na tablicę struktur i jej wielkość;
  - elementy tablicy (wskaźniki na struktury) sortowane są zgodnie z algorytmem sortowania bąbelkowego/pęcherzykowego według pola zawierającego składową typu ***int*** w porządku rosnącym, przy czym algorytm sortowania musi być zaimplementowany **w wersji z iteracjami ograniczonymi oraz mechanizmem kończącym sortowanie po stwierdzeniu, że tablica już jest posortowana**.
- *zliczanie znaków* – funkcja:
  - pobiera wskaźnik na tablicę struktur oraz jej wielkość oraz znak, którego liczbę wystąpień należy wyznaczyć;
  - przeszukuje kolejno struktury w poszukiwaniu zadanego znaku i w przypadku jego znalezienia zwiększa licznik wystąpień o 1.
  - zwraca liczbę wystąpień znaku.

Program po uruchomieniu wczytuje plik wejściowy `inlab01.txt`
Plik `inlab01.txt` zawiera kolejno (rozdzielone spacją): liczbę struktur **N** do wylosowania z zakresu od **1** do **10000** i znak **X** do wyszukania i wyznaczenia liczby jego wystąpień.
Następnie wywoływana jest sekwencja funkcji (dalej w funkcji main())

- czas start;
- losowanie **N** elementów;
- sortowanie;
- zliczanie znaków **X**;
- kasowanie;
- czas stop.

Program wypisuje na konsoli listę **pierwszych 20-tu posortowanych struktur**, liczbę wyszukanych znaków **X** oraz czas wykonania całego programu.

Przygotowanie e-maila do wysłania:

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

- plik z kodem źródłowym musi mieć nazwę: `nr_albumu.sdizo.lab01.main.c` (np. `123456.sdizo.lab01.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: `SDIZO I1 XXXY LAB01`, gdzie `XXXY` to numer grupy (np. `SDIZO I1 210C LAB01`);
- 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).

Wskazówki

Pomiar czasu (wymaga time.h)

```
clock_t begin, end;
double time_spent;
begin = clock();
/* here, do your time-consuming job */
end = clock();
time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```

Wczytywanie z pliku

```
int N;
char X;
FILE* fp = fopen("inlab01.txt", "r");
if (fp == NULL)
return -1;
fscanf (fp, "%d %c", &N, &X);
fclose(fp);
```