Laboratoria nr 3

Podprogramy, rekursja i struktury w C

Przed uczestnictwem w zajęciach student powinien:

- podstawowe wiadomości o rekordach (struktury, unie), tablice rekordów,
- definiowanie typów za pomocą deklaracji `typedef`,
- definiowanie funkcji o zmiennej liczbie parametrów,
- dynamiczny przydział pamięci,
- operacje dyskowe.

Materiały

- `http://www.cs.put.poznan.pl/arybarczyk/C_W_1.PDF`
- `http://www.cs.put.poznan.pl/arybarczyk/C_W_2.PDF`
- `http://www.cs.put.poznan.pl/arybarczyk/C_W_3.PDF`
- `http://www.cs.put.poznan.pl/arybarczyk/C_W_4.PDF`
- `http://www.cs.put.poznan.pl/arybarczyk/C_W_5.PDF`

Zadania

1. Opracować program prowadzący spis pracowników firmy (max. 10 pracowników). Każdy pracownik opisany jest za pomocą struktury zawierającej nazwisko, pensję i procent premii. Program realizuje następujące polecenia:
   - N : nowy pracownik - wczytać dane opisujące i wprowadzić do kolejnej pozycji tabeli struktur,
   - P : nowa wartość pensji dla pracownika o podanym nazwisku,
   - R : nowa wartość procentu premii dla pracownika o podanym nazwisku,
   - W : wypłata, obliczyć ile potrzeba złotych na wszystkie pensje i premie,
   - K : koniec programu.

   Dla realizacji poleceń N, P, R, W zdefiniować funkcje.
2. Zdefiniować unię o nazwie **Bag**, zawierającą pola: **int**, **char**, **float**, służącą do przechowywania odpowiednio: liczb całkowitych, znaków oraz liczb rzeczywistych. Przygotować tablicę **n** losowych elementów typu **Bag**, losowanie powinno obejmować zarówno wybór typu pola jak i jego wartość.
3. Zdefiniować strukturę o nazwie **Student** z polami: **imie**, **nazwisko**, **nr_albumu**, **ocena**. Napisać prosty program, który umożliwi użytkownikowi:
   a. wprowadzenie i wyświetlenie danych grupy **n** studentów – dane należy przechowywać w pamięci wykorzystując tablicę,
   b. modyfikację danych wybranego studenta,
   c. wyznaczenie średniej, maksymalnej i minimalnej oceny dla grupy studentów,
   d. usunięcie danych o grupie studentów.
4. Napisz program będący katalogiem płyt audio (dla maksymalnie 100 płyt). Każda płyta powinna być opisana zestawem danych (tytuł, autor, rok wydania, numer katalogowy, rodzaj muzyki) zawartych w strukturze `Plyta`.
   - Napisz funkcje wczytującą informacje o płycie i wyświetlającą te informacje:
     ```
     Plyta wczytajPlyte();
     void wyswietlPlyte(Plyta obiekt);
     ```
   - Skonstruuj menu w programie umożliwiające wczytanie nowej płyty lub wyświetlenie wszystkich wprowadzonych płyt na ekranie w odpowiednio sformatowanych kolumnach.

- Rozszerz program o podmenu dla wyświetlania płytoteki dotyczące wyświetlania płyt według zadanej kolejności. Dopisz procedurę sortowania płyt na podstawie wybranego elementu struktury, np.:
  ```
  void sortujPlyty(Plyta katalog[100], int parametr);
  ```
  Przydatna do tego celu będzie funkcja porównująca dwie płyty:
  ```
  int porównaj(Plyta pl1, Plyta pl2);
  ```

5. Napisać funkcję potęgowania liczby naturalnej za pomocą mechanizmu wielokrotnego dodawania (rekurencyjnie).
6. Napisać funkcję wyznaczania silni liczby naturalnej *n* (rekurencyjnie).
7. Napisz funkcję, która wypisze w odwrotnej kolejności znaki wczytane z klawiatury (rekurencyjnie). Przykładowo, gdy zadany ciąg to: [1,2,3,a,c,b,\n] program powinien wypisać: [b,c,a,3,2,1].
8. Liczba automorficzna to liczba, która znajduje się na końcu swego kwadratu, np.: $5^2 = 25$, $25^2 = 625$. Napisz program znajdowania wszystkich liczb automorficznych w podanym przedziale domkniętym <a, b>.
9. Napisz program obliczania sumy cyfr dziesiętnych podanej liczby *n* (rekurencyjnie).
10. Napisać program pozwalający na zmianę reprezentacji liczb (dziesiętną, binarną, ósemkową, szesnastkową).
11. Implementacja następujących struktur danych: stos, lista jednokierunkowa, dwukierunkowa, drzewo BST – dostępne powinny być operacje: dodawania, usuwania i przeszukiwania.
12. Napisz program scalania ciągów, który polega na łączeniu posortowanych ciągów w jeden ciąg posortowany.
13. Napisać program pozwalający na obliczanie wartości wyrażeń zapisanych za pomocą odwrotnej notacji polskiej *ONP* (np.: +(-(3,2),1)=2).