Laboratorium – 08.06.2021

Wstęp o strukturach

Struktury to specjalne typy danych, które mogą przechowywać więcej niż jeden typ danych. Rozważmy przykład:

```
struct osoba{
        char imie[6];
        char nazwisko[10];
        unsigned int wiek;
};
```

Stworzyliśmy strukturę *osoba*, w której przechowujemy dane personalne. Taki typ danych może być przechowywany w tablicy osób uczęszczających na dane zajęcia. O wiele wygodniej jest przechowywać takie dane w jednej tablicy, niż tworzyć szereg tablic, z których każda jest poświęcona tylko jednemu typowi danych.

Przypisanie wartości do zmiennych odbywa się poprzez operator '.'. Przykład programu:

```
 struct osoba{
        char * imie;
        char * nazwisko;
        unsigned int wiek;
};
int main(){
        struct osoba pierwsza;
        pierwsza.imie="Ala";
        pierwsza.wiek=30;
return 0;
}
```

W przykładzie zdefiniowaliśmy *structure osoba*. W funkcji *main* stworzyliśmy instancję *pierwsza* typu struktury osoba. Do zmiennej *imie* instancji *pierwsza* przypisaliśmy wartość „Ala", a do zmiennej *wiek* przypisaliśmy 30.

Zauważcie, że typem przy deklaracji zmiennej typu struktura jest *struct <nazwa_struktury>*

Zadania

1. (**10 pkt**)Napisz program, który stworzy plik tekstowy o nazwie „Moje_dane.txt". Zapisz do niego twoje imię i nazwisko instrukcją *fprintf*. Pamiętaj o zamknięciu pliku.

2. (**30 pkt**)Napisz program, który pobiera dane z pliku tekstowego w trzech formatach: napis, liczba zmiennoprzecinkowa oraz liczba całkowita. Dane dla każdego obiektu przechowywane są w jednej linijce i oddzielone są spacjami. Zapisz dane do jednej tablicy, której typem jest struktura danych.

a) Utwórz strukturę *planeta*, która ma 3 zmienne: łańcuch znakowy *name* o rozmiarze 20, liczbę zmiennoprzecinkową (*float*) *distance* oraz liczbę całkowitą (*unsigned int*) *satelity*
b) Utwórz tablicę o rozmiarze 9, będącą typu utworzonej struktury
c) Otwórz plik *planety.txt*. Sprawdź czy plik został poprawnie otwarty. Jeśli plik nie został dobrze otwarty, niech program wypisze do konsoli komunikat „Plik nie został otwarty"
d) Jeśli plik został otwarty, wczytaj zawartość pliku do utworzonej tablicy, posługując się funkcją *fscanf*.
e) Po wczytaniu zawartości pliku, zamknij go instrukcją *fclose*
f) Wyświetl do konsoli liczbę **satelitów trzeciej** i **piątej planety**. Wyświetl również ich nazwy.

Format pliku „planety.txt": Pierwsza kolumna, to nazwy, druga to odległość od Słońca, trzecia kolumna oznacza liczbę satelitów planety:

Merkury 57.9 0
Wenus 108.2 0
Ziemia 149.6 1
Mars 227.9 2
Jowisz 483.88 63
Saturn 1427 56
Uran 2870 27
Neptun 4497 13
Pluton 5900 3

3. (**60 pkt**) Napisz funkcję *sort_planets*, która posortuje elementy tablicy z poprzedniego zadania **rosnąco**, względem zmiennej wybranej przez użytkownika.
   a) Wybór zmiennej do sortowania odbywa się przez podanie do funkcji liczby 1 – sortowanie alfabetyczne ze względu na nazwę, 2 – sortowanie ze względu na odległość od słońca, 3 – sortowanie ze względu na liczbę satelitów (księżyców)
   b) Do funkcji przekazujemy tablicę planet, rozmiar tablicy oraz liczbę
   c) Sama funkcja jest typu *void*
   d) Sortowanie można przeprowadzić metodą bąbelkową:
      - Jeśli element *i+1* jest mniejszy niż element *i,* to dokonujemy zamiany elementów miejscami. Przechodzimy tak po całej tablicy o rozmiarze *N*. Zakres indeksu w pętli *for* powinien być *i<N-1*, ponieważ w każdym wywołaniu odnosimy się do elementu *i+1*
      - Zamiany elementów miejscami dokonujemy tak długo, aż nie zajdzie żadna zamiana.
   e) Sortowanie łańcuchów tekstowych można wykonać funkcją *strcmp*. Funkcja ta zwraca liczbę >0, jeśli łańcuch pierwszy jest alfabetycznie większy niż łańcuch drugi, liczbę 0, jeśli łańcuchy są równe oraz liczbę <0, jeśli łańcuch pierwszy jest alfabetycznie mniejszy

W main:

a) Wypisz sekwencję planet przed sortowaniem.
b) Wywołaj funkcję *sort_planets* na liście planet, sortując po nazwach planet.
c) Wypisz sekwencję planet po sortowaniu do konsoli

d) Zapisz do pliku „Posortowane_Planety.txt” po posortowaniu. Niech każda z planet będzie zapisana w nowej linii pliku. Prawidłowa kolejność:

Jowisz
Mars
Merkury
Neptun
Pluton
Saturn
Uran
Wenus
Ziemia

e) Nie zapomnij o zamknięciu pliku na końcu programu.