LABORATORIUM PODSTAW PROGRAMOWANIA

LAB 5 PAMIĘĆ DYNAMICZNA, STRUKTURY I UNIE

Pamięć dynamiczna

Pamięć komputera, dostępna dla programu, dzieli się na cztery obszary:

- **kod programu**,
- dane **statyczne** (np. stałe i zmienne globalne programu),
- dane **automatyczne -**
  zmienne tworzone i usuwane automatycznie przez kompilator na tzw. **stosie** (*ang. stack*) np. zmienne lokalne wewnątrz funkcji, pamięć dla zmiennych przydzielana jest w momencie wejścia definicji danej zmiennej w „zasięg widzialności" i zwalniana przy wyjściu z tego zasięgu.
- dane **dynamiczne**
  organizowane przez menadżera-zarządcę pamięci dynamicznej, można je tworzyć i usuwać w dowolnym momencie pracy programu, w pamięci wolnej komputera - na tzw. **stercie** (*ang. heap*)

Za alokację pamięci dla zmiennych dynamicznych odpowiada programista, alokacja pamięci (zabierana z obszaru nazwanego sterta), odbywa się na jego żądanie. Alokacji pamięci dla zmiennych dynamicznych dokonuje się za pomocą funkcji **`malloc`** lub **`calloc`**, w przypadku powodzenia funkcje te zwracają wskaźnik na zaalokowany obszar, a w przypadku niepowodzenia zwracają NULL. Do obowiązków programisty należy pamiętanie o zwalnianiu uprzednio zaalokowanej pamięci za pomocą funkcji **`free`**, pominięcie tego etapu może doprowadzić do wyczerpania sterty. Możliwe jest również zmiana rozmiaru uprzednio przydzielonego obszaru za pomocą funkcji **`realloc.`** Nagłówki wspomnianych funkcji zawarte są w pliku nagłówkowym `alloc.h`:

```
void *calloc(size_t nmeb, size_t size);
void *malloc(size_t size);
void free(void *ptr);
void *realloc(void *ptr, size_t size);
```

Parametry **size** oznaczają rozmiar alokowanej pamięci (programista musi obliczyć niezbędny rozmiar posługując się ewentualnie funkcją **`sizeof()).`**

Struktury i unie

Struktura składa się z jednej lub kilku zmiennych dowolnych typów zgrupowanych pod jedną nazwą. Deklaracja struktury składa się z nagłówka zawierającego słowo kluczowe struct i nazwę oraz umieszczonych w nawiasach klamrowych deklaracji zmiennych nazywanych składowymi. Po nawiasie klamrowym zamykającym można umieścić listę zmiennych rozdzielonych przecinkami. Deklarację kończy obowiązkowy średnik. Odwołanie do składowej struktury odbywa się poprzez podanie nazwy zmiennej typu strukturalnego i nazwy składowej połączonych operatorem "**.**", bądź "**->**" (stosowany w przypadku posiadania wskaźnika do struktury). Nazwa struktury jest nazwą typu, można w związku z tym deklarować zmienne tego typu. Unia jest zmienną złożoną ze składowych różnego typu, podobnie jak struktura, z tą różnicą, że umożliwia jednoczesna przechowywanie tylko jednej ze składowych, jej rozmiar jest równy rozmiarowy największej składowej. Przykład unii i struktury poniżej:

```
union LADOWNOSC {
     float ladunekWTonach;
     int iloscPasazerow;
};
```

```
struct SAMOCHOD {
     char marka[20];
     char VIN[30];
     int iloscOsi;
     union LADOWNOSC ladunek;
};
```

Struktury ani unie nie mogą być argumentami jak i wynikami funkcji (ale można używać wskaźników).

Zadania:

1. Napisz program wczytujący dane n studentów i wyświetlający je na ekranie (Zdefiniuj odpowiednią strukturę (nazwisko, data urodzenia, nr indeksu, oceny z 3 przedmiotów) + typedef, napisz funkcję do wyświetlania dowolnego egzemplarza struktury), dynamiczna alokacja pamięci na n studentów.
2. Rozwiń pkt1 o funkcje:
    a. zapis pod wskazany adres danych n studentów
    b. usuwanie studenta
    c. wyszukiwanie studenta wg nazwiska (strcmp(),strstr(), toupper())
    d. posortowanie studentów alfabetycznie (qsort())
    e. znalezienie studenta, który ma najwyższą ocenę z danego przedmiotu
    f. znalezienie studenta z najwyższą średnią ocen