

## Zajęcia 4: Definicja i wykorzystanie struktury

W ramach nowego rozwiązania, stwórz projekt typu **Pusty projekt** działający w trybie konsoli. Dodaj do niego nowy element i zapisz go jako **główny.c**. W nowododanym pliku zdefiniuj funkcję główną **main**. Zdefiniuj w nim strukturę **Ułamek** o następujących składowych:

- **int licznik** – zmienna przechowująca wartość licznika ułamka,
- **int mianownik** – zmienna przechowująca wartość mianownika ułamka.

W pliku głównym przed funkcją główną **main** zdefiniuj również funkcję:

**int NWD(int licznik, int mianownik),**

która ma obliczyć a następnie zwrócić największy wspólny dzielnik liczb **licznik** oraz **mianownik**, korzystając z algorytmu Euklidesa.

W zakresie globalnym programu, zdefiniuj funkcje operujące na zmiennych strukturalnych:

- (1) **struct Ułamek pole\_trojkat(struct Ułamek a, struct Ułamek h)** obliczającą pole trójkąta zgodnie ze wzorem:  $P = a * h / 2$ ; licznik oraz mianownik pola trójkąta mają być liczone oddzielnie, a następnie skracane przez ich największy wspólny dzielnik.
- (2) **struct Ułamek obwod\_trojkat(struct Ułamek a, struct Ułamek b, struct Ułamek c)** obliczającą obwód trójkąta zgodnie ze wzorem:  $O = a + b + c$ ; licznik oraz mianownik obwodu trójkąta mają być liczone oddzielnie, a następnie skracane przez ich największy wspólny dzielnik.
- (3) **int czy\_trojkat(struct Ułamek a, struct Ułamek b, struct Ułamek c)** sprawdzającą, czy trójkąt o bokach: **a**, **b**, oraz **c** w postaci ułamków spełnia warunek istnienia według wzoru:  
 $a + b > c$  i  $a + c > b$  i  $b + c > a$   
 Jeżeli warunek ten jest spełniony, funkcja ma zwracać wartość **1**; w przeciwnym wypadku **0**.
- (4) **int czy\_trojkat\_rownoboczny(struct Ułamek a, struct Ułamek b, struct Ułamek c)** sprawdzającą, czy trójkąt o bokach: **a**, **b**, oraz **c** jest trójkątem równobocznym. Jeśli wszystkie boki są sobie równe (tj. mają ten sam licznik i mianownik), funkcja ma zwrócić wartość **1**, w przeciwnym wypadku **0**.

**Uwaga:** aby uprościć składnię kodu, można zastosować anonimową strukturę z aliasem zamiast definicji struktury. Należy wówczas użyć słowa kluczowego **typedef** w definicji struktury w następujący sposób:

```
typedef struct { ... } Ułamek;
```

Wówczas w deklaracjach/definicjach funkcji **nie jest wymagane** użycie słowa kluczowego **struct**:

```
Ułamek pole_trojkat(Ułamek a, Ułamek h);
```

natomiast zmienne struktury można tworzyć w prosty sposób:

```
Ułamek a = {2, 3};
```

W funkcji głównej **main** należy zaprezentować działanie wszystkich zdefiniowanych funkcji. Przykładowo, dla zmiennych **a** oraz **h** struktury **Ułamek** o polach:

- **a.licznik = 2** oraz **a.mianownik = 3**,
- **h.licznik = 4** oraz **h.mianownik = 5**,

po wywołaniu funkcji **pole\_trojkat**:

```
Ułamek pole = pole_trojkat(a, h);
```

i wypisaniu licznika i mianownika zmiennej **pole**, na ekranie powinna pojawić się informacja: **P = 4/15**.