

Wojciech Myszka

Laboratorium 11a: Struktury danych: Operacje na ułamkach

2020-02-18 07:53:57 +0100

1. Wprowadzenie

1.1. Ułamki czyli liczby wymierne

Jak wiadomo, współczesne komputery wykonując działania na liczbach typu **float** lub **double** (potocznie zwanych rzeczywistymi) wykonują działania na liczbach wymiernych. Niestety liczby, nie tylko są liczbami z ograniczonego zakresu, ale również o ograniczonej precyzji.

Nawet tak proste ułamki jak $\frac{1}{10}$ (0.1) nie posiadają dokładnej reprezentacji ani **float** ani **double**. To co możemy osiągnąć to:

- 0.10000000149011611938 (**float**)
- 0.100000000000000005551115123126 (**double**)
- 0.10000000000000000001355252716 (**long double**)

Stąd pojawia się chęć prowadzenia obliczeń bez błędów. Na przykład na liczbach wymiernych.

Język C nie zna żadnego typu do przechowywania dokładnych wartości liczb wymiernych.

1.2. Liczby wymierne

Liczby wymierne to wszystkie liczby, które można zapisać w postaci ilorazu dwu liczb całkowitych p i $q \neq 0$: $\frac{p}{q}$. Szczególnym przypadkiem liczb wymiernych są liczby naturalne i całkowite.

Bardzo często, w rozważaniach teoretycznych, liczby wymierne przedstawiane są jako para (p, q) . Zapis ten sugeruje również takie prezentowanie liczb wymiernych w programie komputerowym.

2. Reprezentacja komputerowa

Zadaniem Państwa jest zaproponowanie odpowiedniej reprezentacji komputerowej liczb wymiernych.

Można to zrealizować na wiele sposobów. Niektóre będą bardziej wygodne, inne mniej. Ja sugeruję struktury, ale można też inaczej:

- bez żadnych pomysłów: jedna zmienna przechowuje licznik, druga — mianownik,
- każda liczba wymierna to tablica:
 - dwuelementowa: (licznik i mianownik)
 - trójelementowa (część całkowita, licznik, mianownik)
 - czteroelementowa (znak, część całkowita, licznik, mianownik)
- ...

Jeżeli tylko zadziała, to jest OK.

2.1. Struktura

Struktury to definiowany przez użytkownika (programistę) typ danych (pozwalający na grupowanie wartości różnych typów w jednym).

Język C pozwala na dosyć dowolne definiowanie struktur, deklarowanie zmiennych, nadawanie im wartości początkowych oraz dostęp do poszczególnych pól struktury. Na nic więcej. Na przykład:

1. Definicja:

```
struct Point
{
    int x;
    int y;
};
```

x i y to nazwy składowych (albo elementów) struktury.

2. Deklaracja zmiennych:

```
struct Point a, b, c;
```

deklaruje trzy zmienne: a, b, c typu **struct** Point.

3. Nadawanie wartości początkowych zmiennej:

```
struct Point A=(1, 2);
```

4. Dostęp do elementu struktury:

— nadawania wartości:

```
a.x = 5;
```

— odczyt wartości składowej:

```
int u;  
u = a.y;
```

5. Definicja nowego typu: bardzo pożytecznym poleceniem języka C jest **typedef**:

```
typedef struct Point Point;
```

które deklaruje nowy typ Point jako **struct** Point. Ułatwia to deklarację nowych zmiennych:

```
Point C, D, E;
```

Niestety, struktury w języku C nie pozwalają na wykonywanie żadnych operacji na nich. Czyli nie można dokonywać, na przykład, operacji arytmetycznych. Nie można nawet (poza deklaracją) nadać im wartości inaczej niż nadając wartości poszczególnym składowym. Zatem, jeżeli chcemy zmiennej C (typu Point) nadać wartość (100, 200) Trzeba to zrobić w dwu krokach:

1.

```
C.x = 100;
```

2.

```
C.y = 200;
```

3. Zadania do wykonania

1. Zaproponowanie i przedyskutowanie¹ wybranego sposobu zapisu liczb wymiernych (ułamków); należy pamiętać o problemie zapisu znaku, części całkowitej,...
2. Stworzenie biblioteki funkcji matematycznych realizujących operacje:
 - nadawania wartości ułamkowi (korzystając z wartości licznika i mianownika)²,
 - dodawania,
 - odejmowania,
 - mnożenia,
 - dzielenia,
 - upraszczania
 - ...ułamków.

¹ Oczekuję kilku zdań wyjaśnienia przyjętych założeń oraz ich wad/zalet umieszczonych albo w komentarzach programu albo w osobnym dokumencie.

² A znak?

3. **Przetestowanie** tych funkcji.
4. Dodatkowym bonusem będą funkcje:
 - porównywania ułamków,
 - konwersji ułamków do liczb zmiennoprzecinkowych **float/double**
 - ładnego wyprowadzania ułamków; przez ładne uważam coś takiego, co liczbę $-\frac{33}{12}$ wyprowadzi jako:

$$-2\frac{3}{4}$$
 - konwersję liczby zmiennoprzecinkowej **float/double** do ułamka

4. Wersja PDF tego dokumentu...

... pod adresem.

Wersja: 55 z **drobnymi modyfikacjami!** data ostatniej modyfikacji 2020-02-18 07:53:57 +0100