

# Zadanie 1: Arytmetyka

**Opened:** Monday, 16 October 2023, 8:00 AM

**Due:** Wednesday, 8 November 2023, 4:00 PM

## Arytmetyka przybliżonych wartości

Tam gdzie dokonujemy pomiarów wielkości fizycznych, wyniki są obarczone pewnym błędem, np.  $5m \pm 10\%$ . Każdą taką przybliżoną wartość traktujemy jak zbiór możliwych wartości. Zaimplementuj pakiet operacji arytmetycznych na takich przybliżonych wartościach zgodny z załączonym interfejsem [ary.h](#). W szczególności, pakiet ma zawierać:

- definicję typu `struct wartosc`,
- konstruktory:
  - `wartosc_dokladnosc(x, p)` zwraca  $x \pm p\%$  (dla  $p > 0$ ),
  - `wartosc_od_do(x, y)` zwraca  $(x+y)/2 \pm (y-x)/2$  (dla  $x < y$ ),
  - `wartosc_dokladna(x)` zwraca  $x \pm 0$
- selektory:
  - `in_wartosc(x, y)`  $\Leftrightarrow$  wartość  $x$  może być równa  $y$ ,
  - `min_wartosc(x)` = kres dolny możliwych wartości  $x$  (lub  $-\infty$  jeśli możliwe wartości  $x$  nie są ograniczone od dołu, lub `nan` jeśli  $x$  jest puste),
  - `max_wartosc(x)` = kres górny możliwych wartości  $x$  (lub  $\infty$  jeśli możliwe wartości  $x$  nie są ograniczone od góry, lub `nan` jeśli  $x$  jest puste),
  - `sr_wartosc(x)` = średnia (arytmetyczna) wartości `min_wartosc(x)` i `max_wartosc(x)` (lub `nan` jeśli `min_wartosc(x)` i `max_wartosc(x)` nie są skończone),
- modyfikatory:
  - `plus(a, b) = \{x + y : in_wartosc(a, x) \wedge in_wartosc(b, y)\}`,
  - `minus(a, b) = \{x - y : in_wartosc(a, x) \wedge in_wartosc(b, y)\}`,
  - `razy(a, b) = \{x \cdot y : in_wartosc(a, x) \wedge in_wartosc(b, y)\}`,
  - `podzielic(a,b) = \{x/y : in_wartosc(a, x) \wedge in_wartosc(b, y)\}`.

Zakładamy przy tym implícite, że wszystkie argumenty typu `double` są liczbami rzeczywistymi (tzn. są różne od `HUGE_VAL`, `-HUGE_VAL` i `NAN`).

Natomiast w przypadku, gdy wynik nie jest liczbą rzeczywistą, powinien być odpowiednią z wartości: `HUGE_VAL`, `-HUGE_VAL` lub `NAN`.

Rozwiązując to zadanie możesz przyjąć następujące zasady ułatwiające rozumowanie:

- Przyjmij, że modyfikatory domykają wynikowe zbiory wartości – to znaczy, jeżeli wynikiem jest przedział otwarty, to przyjmij, że zostaje on zamieniony na przedział domknięty.
- Operacje na wartościach przybliżonych są monotoniczne ze względu na zawieranie się zbiorów możliwych wartości.  
To znaczy, jeżeli wartości przybliżone  $x, y$  i  $z$  spełniają, jako zbiory możliwych wartości,  $x \subseteq y$ , to:

?

```
plus(x, z) ⊆ plus(y, z)

plus(z, x) ⊆ plus(z, y)
```

- i podobnie dla innych operacji arytmetycznych.
- Kilka przykładów opartych o powyższą zasadę możesz znaleźć w pliku [przyklad.c](#). Komenda kompilacji (pliki [ary.c](#) i [ary.h](#) muszą być w katalogu):  
`gcc @opcje przyklad.c ary.c -o przyklad.e -lm`  
Opcja `-lm` łączy `math.h`. Koniecznie musi znajdować się na końcu komendy kompilacji.
  - Liczby zmiennopozycyjne i operacje na nich potrafią być zaskakujące. Na przykład, standard IEEE przewiduje dwie reprezentacje zera (`+0.0` i `-0.0`), przy czym `1.0 /. 0.0 = HUGE_VAL`, oraz `1.0 /. (-0.0) = -HUGE_VAL`. Może być to pomocne, np. jeśli dzielisz przez wartość przybliżoną, która zawiera jednostronne otoczenie zera. Ale może też okazać się pułapką, gdy rozważasz dzielenie przez wartość dokładnie równą zero. Pamiętaj, że w definicji operacji `podzielic` występuje dzielenie "matematyczne", które nie jest określone gdy dzielimy przez zero. Możesz przyjąć, że liczba typu `double` jest równa zero, jeśli poniższa funkcja daje dla niej wynik `true`:  

```
bool iszero(double x) { return fabs(x) < 1e-10; }
```

Twoje rozwiązanie ma być umieszczone w pliku o nazwie `ary.c` i pasować do specyfikacji interfejsu [ary.h](#). Należy również załączyć plik `ary.h` uzupełniony jedynie o definicję typu `struct wartosc`.

 [ary.h](#)

 [opcje](#)



 [przyklad.c](#)

16 October 2023, 2:21 PM

17 September 2023, 1:54 PM

24 October 2023, 11:08 AM

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Graded
Time remaining	Assignment was submitted 6 days 21 hours early
Last modified	Wednesday, 1 November 2023, 6:51 PM
File submissions	<div><div> <a href="#">ary.c</a></div><div>30 October 2023, 10:07 PM</div></div> <div><div> <a href="#">ary.h</a></div><div>30 October 2023, 10:07 PM</div></div>
Submission comments	<div>► <a href="#">Comments (1)</a></div>

Feedback

Grade	5.00 / 5.00
Graded on	Saturday, 4 November 2023, 3:13 PM
Graded by	 Lukasz Bożyk

Contact us



Follow us

 Contact site support

You are logged in as Witold Formański (Log out)

Data retention summary

Get the mobile app

Get the mobile app

This theme was developed by

conecti.me

Moodle, 4.1.10 (Build: 20240422) | [moodle@mimuw.edu.pl](mailto:moodle@mimuw.edu.pl)