

Zadanie 1.1

C

- Utwórz w katalogu domowych (ikar) podkatalog **jipp**.
- Pobierz plik [trojmian.c](#) i zapisz do w katalogu jipp.
- Otwórz plik w edytorze gedit i przeanalizuj jego zawartość.
- Uruchom terminal, przejdź do katalogu jipp i skompiluj program.
- Uzupełnij kod tak, aby wartość dla parametru *a* były wczytywana w pętli tak długo, dopóki nie zostanie podana wartość różna od zera oraz by w przypadku, gdy wyróżnik (delta) jest ujemny, program wypisywał komunikat o braku pierwiastków rzeczywistych.
- Ponownie skompiluj i przetestuj działanie programu.

Użycie kompilatora gcc:

```
gcc plik.c -o plik
```

Dodatkowo można wykorzystać opcje **-Wall** i **-ansi**, np.:

```
gcc -Wall -ansi plik.c -o plik
```

Jeżeli program używa szczególne biblioteki, to należy dodać opcję dla linkera postaci **-lbiblioteka** (np. **-lm** dla biblioteki matematycznej):

```
gcc -Wall -ansi plik.c -lm -o plik
```

Zadanie 1.2

C

(*) Pobierz plik [pliki.c](#), przeanalizuj jego zawartość, skompiluj i przetestuj jego działanie.

Zadanie 1.3

Java

- Pobierz plik [trojmian.java](#) i zapisz do w katalogu jipp jako plik **Trojmian.java**.
- Otwórz plik w edytorze gedit i przeanalizuj jego zawartość.
- Skompiluj program i uruchom program korzystając z terminala.

Użycie kompilatora javac:

```
javac Plik.java
```

Uruchomienie programu

```
java Plik
```

Zadanie 1.4

Java

(*)

- Uruchom środowisko NetBeans.
- Utwórz nowy projekt (Java → Java Application).
- Uzupełnij funkcję **main** o kod, w którym tworzona jest 10-elementowa tablica liczb całkowitych, wypełniana jest liczbami losowymi z zakresu 1..6 i jej zawartość jest wyświetlana na terminalu.
- Skompiluj i przetestuj działanie aplikacji.

Zadanie 1.5

Java

(*) Napisz aplikację realizującą te same obliczenia co aplikacja z zadania 1.2, ale używając język Java.

Zadanie 1.6

Haskell

Uruchom interpreter języka Haskell (polecenie **ghci**). Wyznacz wartości następujących wyrażeń:

1. $2+3*6$
2. $(12+6)*(7-2)$
3. $\text{sqrt } 3$
4. $\text{sqrt } (12+4)$
5. $\text{gcd } 12\ 8$
6. $\text{rem } 12\ 5$

Zakończ pracę z interpreterem (polecenie **:quit** lub **:q**).

Zadanie 1.7

Haskell

Pobierz plik [silnia.hs](#), przeanalizuj jego zawartość, skompiluj i przetestuj działanie.

Użycie kompilatora ghc:

```
ghc --make plik.hs -o plik
```