

Cel:

- Opanowanie podstaw stosowania konstrukcji sterujących *if* i *switch* w C.

Zajęcia:

1. Utworzenie katalogu roboczego *lab\_5*
2. Utworzenie podkatalogu *switch*
3. Skopiowanie programu *simple\_switch.c* ze strony przedmiotu do podkatalogu
4. Kompilacja w terminalu i sprawdzenie działania
  - Uwaga 1: program zawiera pętlę nieskończoną – zakończenie programu jest możliwe przez przesłanie sygnału zabicia procesu CTRL+C
  - Uwaga 2: w języku C wczytywanie pojedynczego znaku nie odpowiada prostemu wzorcowi "%c" – pozostaje znak odpowiadający klawiszowi *Enter* ( $\backslash n$ , *EOL* - *end of line*). Jednym z możliwych rozwiązań zastosowanym w czytaniu nieskończonej pętli menu w programie *simple\_switch.c* jest użycie wzorca
    - "%c" zawierającego dodatkowy odstęp (spację).
5. Zamiana konstrukcji *switch* na postać standardową – każdy wciśnięty klawisz wywołuje tylko jeden zestaw operacji
  - wprowadzenie koniecznych zmian, w tym modyfikacja komunikatów
  - przetestowanie działania i obserwacja działania instrukcji *break*
6. Napisanie nowego programu o identycznej funkcjonalności jak program z konstrukcją *switch*, ale korzystającego z konstrukcji *if ... elseif ... else*
  - sprawdzenie działania
7. W obu wersjach (*switch* i *if...elseif...else*) zakończenie programu poprzez wciśnięcie danego klawisza
  - rozszerzenie menu, wprowadzenie nowego wariantu wyboru i odpowiadającej instrukcji *break*, komentarze itp.
8. Utworzenie podkatalogu *rownanie\_kwadratowe*
9. Skopiowanie ze strony przedmiotu plików: *rownanie\_kwadratowe.c*, *kompiluj.sh*
10. Analiza pliku *kompiluj.sh*, uruchomienie – wykonanie kompilacji i uruchomienie programu rozwiązującego równanie kwadratowe
  - uwaga na prawa dostępu dla plików tekstowych i wykonywalnych
11. **Uzupełnienie programu tak, aby przed zakończeniem sprawdzał poprawność wyniku – czy obliczone pierwiastki są rzeczywiście pierwiastkami ( $ax^2 + bx + c = 0$ )**
  - **nie należy porównywać z 0, ale wprowadzić parametr TOLERANCJA i sprawdzać jak blisko 0 jest wynik (funkcja *fabs*)**
  - w przypadku kiedy różnica wartości przekracza wartość parametru TOLERANCJA należy umieścić w kodzie odpowiedni napis
12. Analiza wyników:
  - w programie parametry *a*, *b* i *c* są liczbami zmiennoprzecinkowymi (float lub double), nie należy posługiwać się w programie równością zmiennych takich typów, ale ich bliskością (względną lub bezwzględną)
  - należy rozważyć przypadki kiedy *a* jest bardzo małą liczbą, np.  $a=10^{-k}$  dla  $k=5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$ , a *b* i *c* są standardowe, np.  $b = 2, c = 1$  (**liczą się relatywne wielkości współczynników**)
    - jeden z pierwiastków jest zawsze bliski 0.5
    - im mniejsza wartość *a*, tym pierwiastek powinien być bliższy 0.5
    - jednak skończona dokładność obliczeń (błąd zaokrągleń przy dzieleniu przez bardzo małą liczbę) coraz bardziej zaburza wynik dla coraz mniejszych *a* (sprawozdanie

powinno zawierać wartości pierwiastków dla kolejnych wartości  $k$  – osobno dla zmiennych typu float i zmiennych typu double)

- na podstawie przeprowadzonej analizy dobrać parametru TOLERANCJA i, zamiast warunku  $if(a==0)\{...\}$ , wprowadzenie warunku  $if(fabs(a)<TOLERANCJA)$  (zastosowane rozwiązanie jest tylko przybliżone – dla dokładności obliczeń liczą się względne wartości parametrów)

----- 3.0 -----

13. Zapis za pomocą konstrukcji *if, else, elseif* pierwotnej wersji programu *simple\_switch.c*

14. Modyfikacja programu rozwiązywania równania kwadratowego tak, aby rozwiązywał (z odpowiednimi komentarzami) przypadki równania liniowego ( $fabs(a)<TOLERANCJA$ ) i o pierwiastkach zespolonych ( $\Delta < 0$ )

----- 4.0 -----

15. Modyfikacja programu *rownanie\_kwadratowe.c* dla różnych przypadków kontraktu uwzględniającego skończoną precyzję obliczeń

- uwzględnienie w kontrakcie przypadków relatywnie małych wartości  $a$
- uwzględnienie przypadku:  $a$  i  $c$  standardowe (np.  $a=1, c=1$ ),  $b$  duże, np.  $b=10^k$  dla  $k=2, 3, 4, 5, 6, 7, 8, 9, 10$  (jeden z pierwiastków powinien zbliżać się do  $-1/b$ , a drugi do  $b$ )
  - analiza wyników (obliczenia i sprawdzenie poprawności wyniku – dlaczego w pewnym momencie wyniki zaczynają odbiegać od poprawnych?)
  - modyfikacja algorytmu na podstawie materiału z wykładów
  - dla obu przypadków tabela z wynikami dla różnych wartości  $k$  i typów *float* i *double*
- Modyfikacja programu *rownanie\_kwadratowe.c* (najlepiej w nowym pliku, np. *rownanie\_kwadratowe\_nieczytelne.c*) tak, aby wszystkie przypadki były uwzględnione w jednej konstrukcji *if ... elsif ... elsif ... else* (wcięcie ma tylko jeden poziom, poprawność wymaga powtarzania pewnych operacji w różnych gałęziach) – czy taki kod jest czytelny?

Dowolne zadania polecane przez prowadzących

----- 5.0 -----

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie co najmniej kroków 1-12
2. Oddanie sprawozdania o treści i formie zgodnej z regulaminem ćwiczeń laboratoryjnych