# Wstęp do programowania w języku C

Lista zadań nr 8

Na zajęcia 11 grudnia 2017

**UWAGA!** Należy regularnie poświęcać czas na naukę funkcji biblioteki standardowej języka C!

Idea rozwiązania powinna być elegancka i prosta do zrozumienia, a kod czytelny, tj.:

- sformatowany[1] zgodnie z wybraną konwencją,
- zmienne i procedury powinny być nazwane zgodnie z ich przeznaczeniem,
- należy unikać powtarzającego się kodu poprzez zamykanie go w procedury,
- złożone zadania trzeba podzielić na podprocedury.

Dodatkowo swoje programy należy kompilować z flagami «`-g -O -std=c11 -Wall -Wextra`», traktując ostrzeżenia kompilatora jako potencjalne błędy! Programy powinny być wolne od błędów związanych z niewłaściwym użyciem pamięci. W tym celu należy użyć narzędzia instrumentacji kodu AddressSanitizer[2] przez dodanie flagi «`-fsanitize=address`» do opcji kompilatora.

Do każdego zadania należy dostarczyć osobny plik `Makefile` będący instrukcjami dla programu GNU Make[3]. Wywołanie polecenia «`make`» w katalogu projektu musi utworzyć plik wykonywalny; «`make clean`» musi wyczyścić zawartość katalogu, tak by zostały w nim tylko pliki źródłowe.

**Zadanie 1 (15/10).** Napisz grę Wąż[4] (ang. *Snake*) posługując się biblioteką `ncurses` do kontroli terminala tekstowego. Zadanie jest podzielone na etapy. By uzyskać maksymalną liczbę punktów za zadanie pierwsze dwa etapy muszą być ukończone na zajęciach.

**Etap 1.** Posługując się podręcznikiem NCURSES Programming HOWTO[5] skompiluj przykładowe programy z rozdziałów 2.1, 4.7 i 9.2. Zaprezentowano w nich funkcje biblioteki, które przydadzą się w realizacji pozostałych etapów zadania. Napisz «`Makefile`» do budowania programu używającego biblioteki `ncurses` – użyj programu `pkg-config`[6] do ustalenia zmiennych `CFLAGS` i `LDFLAGS`.

**Etap 2.** Zaprogramuj węża sterowanego klawiszami kursora. Początkowo bohater gry może mieć 10 segmentów. Głowa węża przesuwa się o jedno pole w kierunku wybranym przez gracza co ustaloną jednostkę czasu, np. pół sekundy. Jeśli głowa uderzy w ciało węża gra ma się zakończyć.

**Etap 3.** Do gry dodaj jedzenie, które wąż może zbierać, aby urosnąć o jeden segment. Gdy wąż zje pięć jednostek jedzenia z jego ogona wydobywa się jajo. Jedzenie pozostawione na planszy gnije i po losowym czasie znika. Dodatkowo węża należy zamknąć w terrarium, czyli wyznaczyć granice planszy rozgrywki. Jeśli głowa węża uderzy w jego ciało, jajo lub szybę terrarium gra ma się kończyć.

**Etap 4.** Wprowadź planszę tytułową do gry, gdzie będzie można wybrać poziom trudności – im wyższy tym wąż prędzej się przemieszcza, a jedzenie szybciej gnije. Dla każdej rozgrywki wyznacz wynik (ang. *score*). Przydziel pewną liczbę punktów za każdy krok węża, za każde złożone jajo i za każde zebrane jedzenie. Po zakończeniu gry podaj wynik.

[1] `https://clang.llvm.org/docs/ClangFormatStyleOptions.html\#configurable-format-style-options`
[2] `https://github.com/google/sanitizers/wiki/AddressSanitizer`
[3] `http://www.schacherer.de/frank/technology/tools/make.html`
[4] `https://pl.wikipedia.org/wiki/W\%C4\%85\%C5\%BC_(gra_komputerowa)`
[5] `http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/`
[6] `https://people.freedesktop.org/~dbn/pkg-config-guide.html\#faq`

**Zadanie 2 (15).** Napisać program, który rozwiązuje zadanie oznaczone jako *Lista 8 zadanie 2* w systemie Moodle. Rozwiązanie tego zadania będzie sprawdzane automatycznie z użyciem sprawdzarki.