**Lista 6 (Lab) Termin wysłania na**[**SVN**](http://pl.wikipedia.org/wiki/Subversion)**do 7.01.2018**

1. *(5pt)* Wykorzystując program z wykładu, napisz program w asemblerze dla procesora 6502, który wyświetla na ekranie liczbę 8-bitową w kodzie szesnastkowym. Wytłumacz dokładnie zasadę jego działania. Pokaż jak program wykonuje się w emulatorze **emu6502**. Pokaż również jak można wykorzystać opcje **-v** oraz **-m** emulatora.
2. *(10pt)* Wykorzystując program do konwersji BCD z wykładu, napisz program w asemblerze dla procesora 6502, który wyświetla na ekranie liczbę 8-bitową w kodzie dziesiętnym. Wytłumacz dokładnie zasadę jego działania.
3. *(15pt)\** Wzorując się na szybkim mnożeniu dla procesora 6502 przedstawionym na wykładzie, napisz program, który wykonuje dzielenie z resztą liczby szesnastobitowej przez ośmiobitową (czyli wraca wynik dzielenia oraz resztę). Staraj się zrobić dzielenie o najmniejszej złożoności obliczeniowej. Wskazówka: skorzystaj m. in. z instrukcji **LSR**, **ROL** i **SBC**
4. *(15pt)\** Przeanalizuj prosty metaprocesor napisany w 6502 przez Steve'a "Woz" Wozniak'a nazwany [Sweet 16](http://www.6502.org/source/interpreters/sweet16.htm) ([książka](http://www.digibarn.com/collections/books/woz-wonderbook/Apple2WozWonderbook1977.pdf), [artykuły i listingi](http://www.easy68k.com/paulrsm/6502/)). Sweet16 jest pewnego rodzaju maszyną wirtualną, który pozwala na wykonywanie przez 8 bitowy procesor 6502 prostych instrukcji 16 bitowych. Zaimplementuj (przepisz) na **xasm** oraz napisz prosty program demonstrujący jego działanie. Następnie zamiast instrukcji **LDD** oraz **STD** wprowadź instrukcje przesunięcia bitowego **SHL** (SHift Left) mnożenie przez 2 oraz **SHR** (Shift Right) dzielenie przez 2 np. **SHL Rn** mnoży przez 2 zawartość rejestru **Rn**.
5. *(5pt)* Napisz program w asemblerze x86 (32-bit), który wysyła na standardowe wyjście [liczby bliźniacze](https://pl.wikipedia.org/wiki/Liczby_bli%C5%BAniacze) mniejsze od 10 000.
6. *(10pt)* Napisz program w asemblerze x86 (32-bit), który liczy silnię podwójną n!! na liczbach 128 bitowych. Liczba n pobierana jest z argumentów **argc, argv** (lub ze standardowego wejścia) oraz wynik wysyłany jest na standardowe wyjście.
7. *(10pt)* Napisz w asemblerze x86 (32-bit) program do obliczania wyrażeń arytmetycznych wprowadzonych w odwrotnej notacji polskiej z wykorzystaniem koprocesora matematycznego. Program pobiera ze standardowego wejścia łańcuch znaków np. **2 3 4 \* +**. Wykorzystując koprocesor matematyczny oblicza wynik (na liczbach zmiennoprzecinkowych) oraz wysyła wynik na standardowe wyjście.
8. *(15pt)\** Napisz funkcje w asemblerze x86 (32-bit) z wykorzystaniem koprocesora matematycznego, które obliczają:
   1. lnxln⁡x
   2. exex
   3. sinhx=ex−e−x2sinh⁡x=ex−e−x2
   4. sinh−1x=ln(x+√x2+1)sinh−1x=ln⁡(x+x2+1)
9. (15pt)\* Napisz program, który wykorzystuje dowolny niskopoziomy dostęp do ekranu np. [framebuffer](http://betteros.org/tut/graphics1.php) lub [SDL](https://www.libsdl.org/) lub [inny](http://wiki.osdev.org/How_do_I_set_a_graphics_mode). Inicjalizacja trybu graficznego może zostać napisana w języku C. Następnie napisz w asemblerze x86 program rysujący [zbiór Mandelbrota](http://en.wikipedia.org/wiki/Mandelbrot_set).
10. *(10pt)* Napisz program w asemblerze dla procesora ARM, który oblicza największy wspólny dzielnik dwóch liczb. Pobieraj dwie liczby, ze standardowego wejścia i wysyłaj wynik na standardowe wyjście.