**1.** Należy stworzyć zestaw funkcji realizujących proste operacje arytmetyczne:

1. **dodawanie,**
2. **odejmowanie,**
3. **mnożenie,**

na wielkich liczbach w kodzie naturalnym, binarnym i/lub U2. Przeprowadzane operacje powinny wykorzystywać wbudowane w procesor sprzętowe mechanizmy ułatwiające wykonywanie takich operacji, np. propagacja przeniesienia pomiędzy kolejnymi pozycjami.

**2.** Poprzez wielkie liczby rozumie się liczby zawierające co najmniej kilkaset bitów. Reprezentacja liczb powinna uwzględniać możliwe różnice w ich długości, tzn. należy unikać reprezentacji o stałej liczbie pozycji. Rozwiązaniem może być np. przechowywanie liczb w strukturze zawierającej liczbę zajmowanych pozycji i tablicę z wartościami poszczególnych cyfr.

**Tips**: adcl, movl, pushl, clc

.data

liczba1:

.long 0x10304008, 0x701100FF, 0x45100020, 0x08570030

liczba2:

.long 0xF040500C, 0x00220026, 0x321000CB, 0x04520031

Podpowiedzi:

* Wyniki poszczególnych operacji arytmetycznych na stos (każda operacja w danej iteracji -> jej wynik na stos np. suma cząstkowa)
* Ostatnie przeniesienie (przepełnienie też na stos)
* GDB - do podglądu zawartości stosu
* makefile z możliwością podglądu symboli w GDB:

dodawanie: dodawanie.o

ld dodawanie.o -o dodawanie

dodawanie.o: dodawanie.s

as **--gstabs** dodawanie.s -o dodawanie.o

* etykieta .code32 na starcie programu (pozwoli na operacje na słowach 32-bitowych)
* praca krokowa w GDB (step) i podgląd całego wyniku na stosie
* JNC - skok jeżeli brak przeniesienia - pozwala na zapis ostatniego przeniesienia w postaci “jedynki” do rejestru i na stos