

## Ćwiczenie nr 4. Kalkulator ONP

Zapoznaj się z odwrotną notacją polską: [https://pl.wikipedia.org/wiki/Odwrotna\\_notacja\\_polska](https://pl.wikipedia.org/wiki/Odwrotna_notacja_polska)

Zaimplementuj prosty kalkulator (liczb całkowitych lub zmiennoprzecinkowych – Twój wybór) oparty o ONP, działający w trybie ciągłym. To znaczy: nie należy próbować parsować całego wyrażenia, program powinien działać w pętli, która co każdą iterację albo pobiera od użytkownika liczbę, albo kod operacji.

Zwróć uwagę, że w MARS nie ma możliwości rozróżnienia między wprowadzeniem przez użytkownika liczby bądź znaku (syscall 5 *wymaga* podania liczby i spowoduje błąd runtime'u MARSa jeśli użytkownik poda np. „x”). Dlatego więc w pętli głównej program powinien:

- drukować obecny stan stosu,
- pytać użytkownika, czy ten chce wprowadzić kolejną liczbę, czy wykonać operację (0/1),
- pytać użytkownika o typ wykonywanej operacji (0 – dodawanie, 1 – odejmowanie, 2 – mnożenie, 3 – dzielenie).

Po każdej operacji wynik powinien być umieszczany z powrotem na stosie.

Pamiętaj: do wykonania operacji należy zdjąć ze stosu dwie liczby. Zabezpiecz program przed sytuacją, gdy użytkownik żąda wykonania operacji, podczas gdy na stosie nie ma dość elementów.

Podział na etapy:

- oprogramowanie protokołu obsługi stosu złożonego z pseudo-funkcji „push”, „pop”, „isEmpty” (2.0 pkt jeśli wykorzystany jest dodatkowy rejestr zliczający obecne elementy na stosie, 2.5 pkt jeśli implementacja nie wykorzystuje zbędnych rejestrów; jeśli masz zamiar porzucić na tym kroku, oprogramuj podstawową pętlę pozwalającą na wczytanie pewnej liczby elementów, a następnie zdejmującą i drukującą wszystkie liczby ze stosu)
- oprogramowanie pętli głównej, pobierającej wartości i wykonującej operacje (3.5 pkt)
- wyświetlanie stanu stosu przy każdej iteracji (4.5 pkt)
- zabezpieczenia (5.0 pkt)