

Systemy Operacyjne

Laboratorium 6 – Lista A

Utwórz skrypt *arithmetic.sh*. Nadaj mu prawo do wykonywania. **Przed przystąpieniem do wykonywania zadań wydaj w terminalu następujące polecenie:**

```
1 $ulimit -u 1000
```

Zapobiegne ono zawieszeniu się systemu przy podaniu nieprawidłowego warunku stopu w funkcji rekurencyjnej.

W skrypcie umieść rozwiązania poniższych zadań (**Przed kodem dotyczącym każdego podzadania umieść komentarz z numerem zadania np. #Z1**):

Zadanie 6.0(Pkt. 3):

Napisz funkcję przyjmującą dwa argumenty **\$1** i **\$2**, zwracającą wartość $\frac{\$1+\$2}{\$1-\$2}$

- Sprawdź czy mianownik jest różny od zera. Jeśli nie, wyświetl komunikat o błędzie oraz przerwij wykonanie skryptu, zakończ skrypt kodem błędu **-1 (exit -1)**.
- Funkcja zwraca wynik poprzez wypisanie go na standardowe wyjście (polecenie **echo**).

Zadanie 6.1(Pkt. 3):

Napisz funkcję obliczającą silnię $n!$ rekurencyjnie.

- Funkcja przyjmuje jeden argument. Sprawdź, czy liczba argumentów jest właściwa. Jeśli nie, wyświetl komunikat o błędzie oraz przerwij wykonanie skryptu, zakończ skrypt kodem błędu **-1 (exit -1)**.
- Funkcja zwraca wynik poprzez wypisanie go na standardowe wyjście (polecenie **echo**).

Zadanie 6.2(Pkt. 3):

Napisz funkcję obliczającą silnię $n!$ iteracyjnie.

- Wykorzystaj pętlę **for**.
- Funkcja przyjmuje jeden argument. Sprawdź, czy liczba argumentów jest właściwa. Jeśli nie, wyświetl komunikat o błędzie oraz przerwij wykonanie skryptu, zakończ skrypt kodem błędu **-1 (exit -1)**.
- Funkcja zwraca wynik poprzez wypisanie go na standardowe wyjście (polecenie **echo**).

Zadanie 6.3(Pkt. 4):

Napisz funkcję obliczającą sumę wszystkich przekazanych argumentów.

- Funkcja przyjmuje dowolną liczbę argumentów (argumenty przechowuje tablica '\$@').
- Funkcja przyjmuje argumenty będące liczbami całkowitymi.
- Funkcja zwraca wynik poprzez wypisanie go na standardowe wyjście (polecenie **echo**).