Zadanie 1.

Napisać moduł `zad1`, który uruchomi `N` nowych procesów, każdy powinien wypisać trzy razy `Tu proces (numer procesu)`. Liczba `N` powinna być argument funkcji `start/1`.

Zadanie 2.

Napisać moduł `poczatek.erl`, który uruchomi dwa procesy `procA` i `procB`. Ich działanie jest następujące: proces A wysyła do procesu B atom `czesc`, natomiast proces B wyświetla informację o jego otrzymaniu. Całość powinna wypisywać na standardowe wyjście następujące informacje

```
Process B started with PID=<0.13.0>.
Process A started with PID=<0.12.0>.
Process A sends atom czesc to B.
Process B received atom czesc.
```

Po wykonaniu powyższej komunikacji, oba procesy powinny zakończyć swoje działanie.

Uwaga: Przydatna może być funkcja `self()`, która zwraca identyfikator wywołującego ją procesu.

Zadanie 3.

Napisać moduł `cs1.erl`, który uruchomi dwa procesy: klienta i serwera. Klient powinien przyjmować jako argument listę liczb całkowitych (oraz PID serwera) i w trakcie działania wysyłać do serwera **po jednym** elemencie tej listy. Serwer powinien zaś każdą liczbę podnosić do kwadratu i wypisywać wynik na standardowe wyjście.
Po przesłaniu ostatniego elementu listy, klient powinien przesłać atom `end_of_list`. Server otrzymując taki atom, powinien zakończyć działanie.

Zadanie 4.

Zmodyfikować poprzedni program tak, aby serwer odsyłał do klienta wynik swoich działań i to klient teraz powinien wyświetlać otrzymaną liczbę. Przykładowy schemat działania:

```
Server started with PID=<0.12.0>
Client sends 10
Client receives 100
Client sends 20
Client receives 400
Server is going down...
```

Dodatkowo serwer powinien kończyć swoje działanie, jeśli żadne żądanie nie przyjdzie przez 3 sekundy (**klient na koniec nie wysyła atomu, który o tym informuje.**)
Moduł powinien nosić nazwę `cs2`.

Zadanie 5 (domowe).

Napisać moduł `types.erl`, który uruchomi dwa procesy: klienta i serwera. Klient ma jeden argument, który przesyła do serwera. Serwer działa zaś w nieskończonej pętli i w zależności od tego, co otrzymał wypisuje na standardowe wyjście:
`Server received number, server received list`. W momencie otrzymania przez serwer atomu `koniec`, kończy on swoje działanie.
Uwaga. Do komunikacji z serwerem proszę nie używać PID'a serwera, ale należy skorzystać z zarejestrowanej nazwy.