**Testarea unor smartcontracte Ethereum**

- Testare structurala –

**Introducere**:

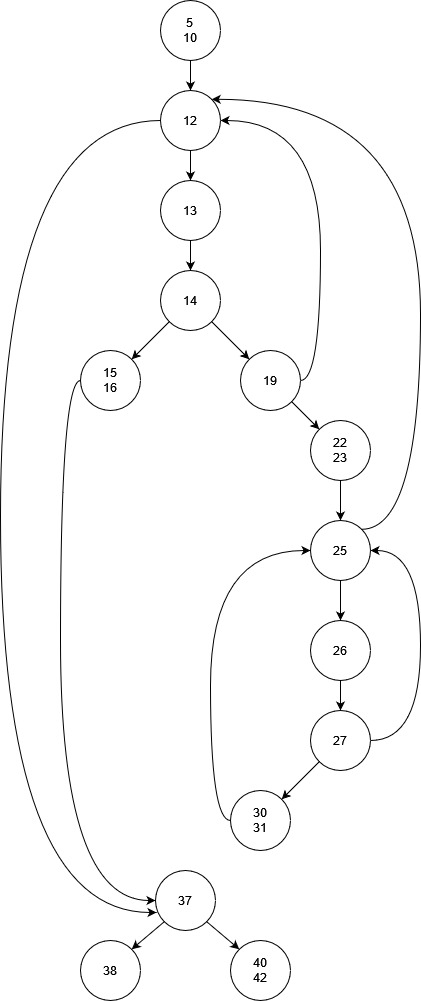
Aplicația pe care o avem de testat implementează prin intermediul mai multor smartcontracte o piață de tokeni ERC20. Utilizatorii pot creea tokeni ERC20, dar și Liquidity Pool-uri (LP) ce funcționează drept o stație de schimb valutar. Practic, un LP conține doi tokeni în anumite cantități, iar utilizatorii pot sa depuna lichiditate (fonduri) în aceste LP-uri, dar șisă le folosească pentru a schimba o anumită cantitate din tokenA în tokenB. Pentru ca acest schimb să se realizeze, trebuie să exite, inițial, un LP care schimbă din tokenA în tokenB. Cum acest lucru nu este mereu posibil, am ales să m (ca număr de LP-uri) pentru schimbul între tokenA și tokenB. Astfel, nu mai este nevoie ca cei doi tokeni să fie “conectați” direct printr-un LP, ci doar să existe un drum între cei doi tokeni in graful descris.

Din contractul LPRouter am ales să realizăm testarea structurală pentru funcția ce calculează drumul minim între cei doi tokeni. Codul aceste funcții se află pe pagina următoare.

Pentru început, am construit Control Flow Graph-ul (CFG) asociat functiei pe care o testam din smartcontract-ul LPRouter. Graful respectiv se află pe pagina de după cea care conține codul funcției.

**Codul funcției de testat:**



**CGF:**