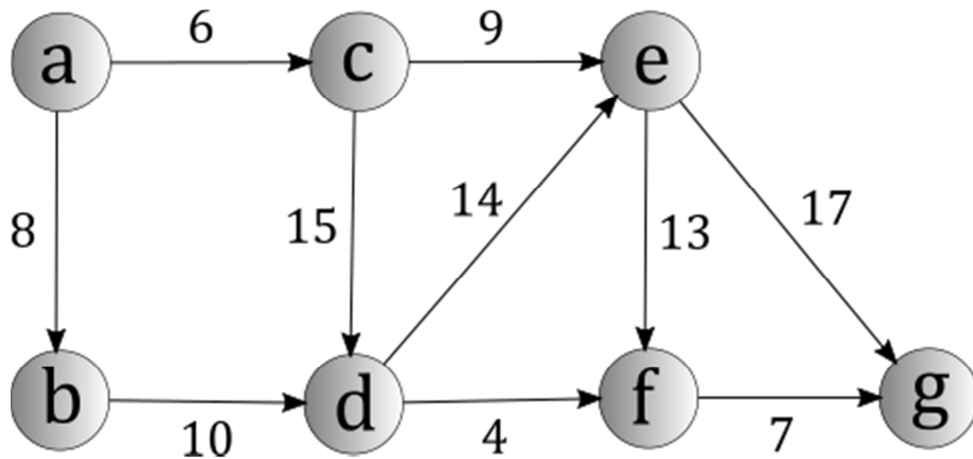


## ZADATAK

1. Napisati funkciju *MakeGraph* koja formira graf sa slike 1 i vraća ga kao povratnu vrednost.  
`def MakeGraph() --> returns graph`



Slika 1 – Primer ispitnog grafa.

2. Napisati funkciju *GetInDegrees* i *GetOutDegrees* koje računaju ulazne i izlazne stepene (rangove) svakog čvora u grafu. Povratna vrednost obe funkcije je lista stepena čvorova koji su raspoređeni u istom redosledu u kome se nalaze u ulaznom grafu.

```
def GetInDegrees(graph) --> returns List  
def GetOutDegrees(graph) --> returns List
```

3. Napisati funkciju *ShortestPath* koja računa najkraću putanju u datom grafu između dva čvora koja se prosleđuju kao parametar. Povratna vrednost je **par vrednosti (najkraća putanja, dužina)**. Koristeći ovu funkciju izračunati najkraću putanju između čvorova A i G sa grafa sa slike iz zadatka (1).

```
def ShortestPath(graph, nodeA, nodeB) --> returns (List, int)
```

4. Napisati funkciju *UpdateEdge* koja dodaje ivicu između dva čvora sa težinom koja se prosleđuje kao ulazni parametar. Ukoliko ivica već postoji između ta dva čvora, ova funkcija menja težinu date ivice.

```
def UpdateEdge(graph, nodeA, nodeB, weight) --> returns None
```

5. Modifikovati graf napravljen u zadatku (1) koristeći funkciju iz zadatka (4) **dodajući ivicu od čvora B do čvora C sa težinom  $w(B, C) = -4$** . Napisati funkciju *NewShortestPath* koja računa najkraću putanju između čvorova A i G u ovom grafu. (Funkcija nema parametre, koristiti globalnu promenljivu koja predstavlja modifikovan graf dobijen kao deo ovog zadatka)

```
def NewShortestPath() --> returns (List, int)
```