

## Uppgift 1 - Tabell av kostnader

Pontus är lite upprörd över att priserna på brädspel gått upp så mycket på senaste tiden, framförallt spel med massa plastfigurer. För att få lite bättre förståelse av detta vill han ha ett program som kan beräkna kostnaden för plast när volymen ökar.

Användaren får mata in startvolym, slutvolym, steglängd och priset på plast. Programmet skriver sedan ut en tabell formaterad som i körexemplet. All utdata skrivs ut högercentrerad och bredden av de två första kolumnerna är sju tecken och den tredje kolumnen är åtta tecken.

Inmatningen från användaren behöver felkontrolleras. Startvolymen behöver vara positiv, slutvolymen behöver vara större än startvolymen, steglängden och priset behöver vara positiva. Om användaren matar in fel behöver nya chanser att mata in korrekt ges tills användaren lyckas. Om användaren matar in en inkorrekt slutvolym skall alltså nya chanser ges att mata in slutvolymen tills den stämmer. Du behöver inte kontrollera att användaren matar in rimliga typer (användaren kommer mata in heltal/flyttal där det förväntas).

### Körexempel (användarinmatning i fetstil):

Mata in startvolym (cm3): **1**

Mata in slutvolym (cm3): **-1**

Mata in slutvolym (cm3): **0**

Mata in slutvolym (cm3): **4**

Mata in steglängd: **0.5**

Mata in priset för plast: **3**

Price Volume Product

=====

3	1.0	3.0
3	1.5	4.5
3	2.0	6.0
3	2.5	7.5
3	3.0	9.0
3	3.5	10.5
3	4.0	12.0



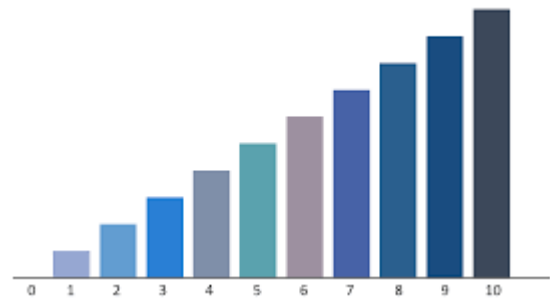
## Uppgift 2 - Räkna ord i en fil

I filen **text.txt** hittar du ett utdrag från hemsidan boardgamegeek. Ditt jobb är att skapa ett program som läser igenom varje ord i denna fil och räknar antalet förekomster av varje ord. Ett ord definieras i denna uppgift som godtyckliga tecken som separeras med mellanslag (regex är överkill för detta problem).

När orden räknats ska du skriva ut de 10 vanligaste förekommande orden enligt körexemplet nedan.

### Körexempel:

```
Board: 106
Game:: 100
Amazon:: 99
the: 91
of: 76
and: 63
List:: 59
to: 52
Game: 44
your: 40
```



## Uppgift 3 - Sammanfoga sammansatta datatyper

I filen **uppgift3.py** finns det ett givet huvudprogram som skapar två **listor** med spel(**dictionaries**) och sedan anropar två funktioner. Ditt jobb är att skapa dessa två funktioner som anropas så att huvudprogrammet producerar förväntat resultat.

Funktionen **display** ska gå igenom alla spel i listan och skriva ut namnet och betyget för varje spel i listan. Ordningen spelar ingen roll.

Funktionen **merge** tar in två listor med spel och sammanfogar dessa. Sättet detta ska fungera på är att en ny lista skapas (de listor som skickas in skall förbli oförändrade). Alla spel som finns med i någon av listorna finnas med i den nya listan. Men om ett spel finns med i båda listorna ska det bara finnas med en gång i den nya listan, men med medelvärdet av betygen från båda listorna.

Ett spel räknas här som samma spel om både **name** och **publisher** är samma.

Tänk på att en sammanfogad lista ska vara en helt ny lista, det är alltså viktigt att du håller koll på att du kopierar spelen som infogas i den nya listan och inte bara infogar referenser till samma spel i den nya listan.

### Körexempel:

bgdb1:

Gloomhaven 9

Decent 8

Codenames 6

bgdb2:

Eclipse 7

Twilight Imperium 8

Codenames 7

bgdb1 & 2 merged:

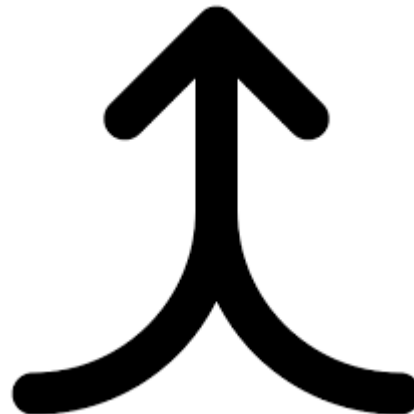
Gloomhaven 9

Decent 8

Codenames 6.5

Eclipse 7

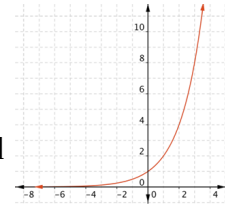
Twilight Imperium 8





## Uppgift 5 – Horisontell tabell

Det visar sig att volymen av en kub växer väldigt snabbt när längden av kubens sidor ökar, det är en så kallad kubisk tillväxt. Detta förklarar varför plastfigurer till brädspel ökar så snabbt i pris när de blir större. Ditt jobb är att i denna uppgift skapa en tabell över volym för en kub med växande längd av sidorna.



I denna uppgift styrs innehållet i tabellen från kommandoraden. Det första argumentet efter filnamnet är startlängden, det andra argumentet steglängden och det tredje argumentet är antal kolumner som ska skrivas ut.

En mindre rimlighetskontroll av kommandoradsargumenten behöver utföras. Vi behöver kontrollera att antalet argument som skickas med stämmer. Alla siffror som skickas in behöver vara positiva. Om en siffra förväntas skickas in kan din felkontroll anta att användaren inte matar in några andra tecken än just siffror.

Den första raden i tabellen är längden av en av kubens sidor. Den andra raden är motsvarande volym för en kub med den storleken. Volymen för en kub beräknas genom  $\text{sida} \times \text{sida} \times \text{sida}$ . Observera att måttenheten för längden är millimeter och volymen är kubikcentimeter. Det går 1000 kubikmillimeter på en kubikcentimeter.

```
python3 uppgift5.py 28 5 7
Height (mm) :      28      33      38      43      48      53      58
Volume (cm3) :      22      36      55      80     111     149     195
```

```
python3 uppgift5.py 28 5
Error: Wrong number of arguments, three numbers are expected.
```

```
python3 uppgift5.py 28 5 -1
Error: Number of columns needs to be a positive number
```