

## Uppgift 1

### Reflektera kring planering och genomförande

Jag planerade min uppgift genom att läsa igenom allt en gång och sedan fokusera på första delen i uppgiften, för att sedan fokusera på nästa del.

Att planera koden stegvis genom tre steg kändes som att det skulle räcka.

De tre stegen innehöll:

- läsa och förstå vad som ska göras
- planering: Tänka igenom vilka metoder som kan användas till koden
- tidsåtgång: Vilken tid det tar att göra eventuell research och implementering.

[1.a]

#### Planeringen

Uppgiften såg väldigt enkel ut. Därför tänkte jag att jag inte behövde planera något utan att allt kommer naturligt genom kodningen. Det enda jag tänkte på i första hand var att hitta en lösning med `split()` metoden.

#### Implementeringen

Jag hittade en lösning väldigt kvickt med `split()` och använde mig utav den. Under tiden kom jag på att det är skillnad på versaler och gemener i programmering, så under tiden tänkte jag ut en lösning med `ToLower()` metoden för att kunna räkna ut både `a` och `A`.

När jag var klar med uppgiften hade jag glömt bort att det ska räknas upp både gemener och versaler för sig. Så jag fick ändra om koden genom att ta bort `ToLower()` metoden och duplicera befintlig kod och ge ett nytt variabel namn som jag ville undvika från början, men det klarade kraven iallafall.

Planerad tid : 30 min

Implementering: 10 min

Testning : 2 min

Sökning : 3 min

Tid att implementera om : 5 min

Total tid 20 min

Orsak till avvikelse.

Att hitta en bra implementation till `split()` hittades på första sökningen.

Uppgiften var simplare än vad jag trodde.

Formulerade frågan fel och räknade ihop både `a` och `A` tillsammans.

[1.b]

### Planeringen

Då `.Split()` funkade bra på förra uppgiften så fortsätter på det spåret och lägger in värdena i en array som jag sorterar ut ojämna och jämna tal med modulus och nollorna får bli genom en if sats

### Implementeringen

Då `Split()` inte gick att använda med heltal så sprack hela min planering och fick tänka om. Jag sökte runt och hittade en bra lösning med lamda uttryck och kunde använda mig av denna lösning hela vägen.

Planerad tid : 45 min

Implementering: 30 min

Testning : 5 min

Sökning : 35 min

Total tid 70 min

Orsak till avvikelse.

Hela min planering sprack då den inte gick att utföra.

Testade på annan lösning men fick inte ihop det med modulus

La ner mycket tid på sökning men tack vare det hittade jag ett bra sätt att implementera koden på.

[1.c]

### Planeringen

Efter en lång planering för att fundera ut hur jag ska lösa uppgiften utan arrayer eller andra datastrukturer kom jag på mig själv att jag behöver ju inte spara talen utan jag ska endast skriva ut det näst största talet.

Så jag kunde inte komma på någon enklare lösning med if sats i en for loop som bara spara de två högsta variablerna.

För att ändra antalet inmatade tal kommer användaren få skriva in valfritt tal som kommer implementeras med en while loop

### Implementeringen

Jag tänkte mig en while loop till en början som såg till att antal inmatade tal returnerat vid inmatat tal. Under implementeringen märkte jag att jag kunde kommentera ut while loopen och koda det mesta i en for loop.

Planerad tid : 45 min

Implementering: 30 min

Testning : 5 min

Sökning : 10 min

Total tid 45 min

Orsak till avvikelse.

På grund av den långa planeringen samt att vi har laborerat liknande uppgift förut kunde jag planera tiden bra.

Jag hade visserligen räknat med en while loop men den tog jag bort