

## Specifikation till uppgift XX

Skriv här en kort beskrivning av vad programmet ska göra.

### Funktioner

Under den här rubriken skriver ni ner vilka funktioner som programmet kommer behöva. Om det är möjligt är det bra att redan här specificera varje funktions parametrar och vad de ska returnera.

Exempel:

```
# den här funktionen summerar alla tal i listan mellan index startIndex  
och slutIndex  
def interval_sum(lista med int:s, startIndex, slutIndex):  
    ...
```

```
# den här funktionen konkatenerar alla strängar i listan  
def concat(lista med str:s):  
    ...
```

```
# den här funktionen använder SL:s API för att returnera det site ID som  
bäst motsvarar söksträngen station  
def getSiteID(station):  
    ...
```

```
# main-funktionen kör programmet  
def main():  
    ...
```

### Datastrukturer

Under den här rubriken skriver ni vilka datastrukturer ni tänker att programmet kan behöva använda sig av. Exempel på datastrukturer är listor, dictionaries, tupplor och tekniskt sett även strängar. Relevant här är både datastrukturer som ni själva introducerar och sådana som "tvingas på er" av API:s eller liknande. Använd gärna bilder för att visualisera hur datastrukturerna ser ut, eftersom det då blir tydligare hur de ska användas.

Exempel:

- "För att använda SL:s API behöver vi hantera JSON-filer"
- "JSON-filer tolkas i Python som antingen listor eller dictionaries"
- "När vi söker efter ett siteID behöver vi en söksträng"
- "Vi kan lagra alla svar vi får från API:t i en lista"

### Algoritmer

Här skriver ni ner algoritmer som kommer behövas i programmet. Det kan till exempel vara en algoritm för att plocka ur en relevant del ur en JSON-fil, en algoritm för att välja ett specifikt element ur en lista eller något annat.

Exempel:

Algoritm för att hitta den sekvens vars summa är störst, från en lista av integers:

1. Initialisera variablerna *start* och *end* till 0 och *max* till värdet av listans första element.
2. Gå igenom varje möjligt startindex *i*, som går från 0 till 1 mindre än längden av listan.
3. För varje möjligt startindex *i*, gå igenom varje möjligt slutindex *j*. *j* kan alltså gå från *i* till 1 mindre än längden av listan.
4. Summera alla tal i intervallet. Om summan är större än *max*, sätt *max* till det nya värdet, *start* till *i* och *end* till *j* - 1.
5. Resultatet finns i variablerna *start* och *end*.

## Tidsplan

Tanken med att göra en tidsplan är att fundera över i vilken ordning saker ska göras och att uppskatta hur lång tid varje steg tar. Det är en bra övning att göra för att kunna bedöma om er projekttid är genomförbar eller inte. Ni kan dela upp tidsplanen i veckor, dagar, eller något annat som ni tycker är relevant.

Exempel:

Vecka 7 (3 h):

Under den här veckan ska vi studera våra API:s och fundera över vilka datastrukturer vi kan behöva använda för att plocka ut relevant data. Vi ska även få GitHub att fungera.

Vecka 8 (3 h):

Den här veckan ska vi skriva funktionerna.....