

# Laboration 1.a-1.c

## Uppgift 1.a:

Till och börja med är det viktigt att jag pekar ut att jag missförstod uppgifts "valet", eller bättre sagt, jag trodde jag hade ett uppgifts val dvs. att jag bara behövde göra en av uppgifterna 1.a 1.b eller 1.c. Då jag började med uppgift 1.b så fastnade jag vid ett ställe mitt i, då tänkte jag att jag kan ta och göra 1.a för att jag kanske då kanske inser hur jag kan jobba runt om problemet jag hade eftersom jag tyckte att uppgifterna var ganska lika. Så jag gjorde den på en ca. 20 min utan planering eller något. Samt så såg jag problemet som jag hade (jag går in mer på det i reflektion av 1.b).

Simpelt sagt så har jag ingen planering eller reflektion att redovisa för att jag hade missförstått uppgiften.

## Uppgift 1.b planering:

Steg 1: läs in ett tal därefter processera en av siffrorna i taget och ge +1 för det område det ska läggas i.

Steg 2: lägg till fel meddelanden samt try-catches vid behov.

Steg 3: granska koden.

Jag misstänker att uppgiften kommer att ta 1½ tim – 2 tim för att få ut det i önskad kvalité.

## Uppgift 1.b reflektion:

**Intro** – Detta gick **inte** som planerat. Jag bestämde mig för att göra 1.b för att jag tyckte den svårighetsgrad inte var allt för svår, att jag då redan kunde allt som krävdes och att jag bara kunde skriva allt i ett svep utan problem.

Jag bestämde mig för att använda språket C# för att inte bli för van med "slappheten" med javascript. Samt så föredrar jag att använda C# framför javascript ett enormt. (Kommer också använda C# för att framöver uppgifter.)

**Problem som uppstod samt lösning** – Jag hade tänkt bara göra en for-loop som för var tecken ger +1 för dess respektive värde, dvs. nolla, jämnt eller ojämnt. Men jag hade bl a. glömt striktheten i C# att man inte kan använda variabler som skapas i te x. en loop utanför den.

Samtidigt fick jag problem med jämföringen av siffrorna, asså när jag tilldelade en variabel value ett värde ifrån input ( value=input[i] ) så visste jag inte att input[i] var av datatypen char, när value var av datatypen int. Så den överföringen funkade inte redigt eftersom jag aldrig int.Parsade. Så jag bestämde mig för att ta en liten luft paus och tänka på det och sen bestämde jag mig att göra 1.a för att kolla om jag kunde inse vad felet var för något. Det var under tiden jag gjorde den som jag insåg att input gav ett värde av datatypen char.

Efter detta så löste resten av problemen upp sig själva och allt jag behövde göra var att lägga till mina fel medelanden och granska koden.

*Tiden* – uppgiften tog 1 ½ tim, om man då räknar med 1.a så blev det runt 2 tim avrundat.

**Slutvärdering** – Jag underskattade verkligen detta. Min gissning på hur lång tid det skulle ta tyckte jag att jag överdrev med, bara för att vara på den säkra sidan. Jag förväntade mig en då att det kanske bara skulle ta 30-60 min max.

Jag har verkligen insett viktigheten med en solid planering eftersom att efter ha haft ett hål i min planering som startade en kedja med flertal andra problem och praktiskt taget förstörde min planering jag hade i huvudet. Jag har inget annat att skylla på än min planering, jo i slutändan så höll jag mig till den relativt bra men det hade säkert gått betydligt bättre om jag bara tänkt igenom min planering mera och förtydligat den. Jag hade inte heller gjort mig redo att möta problem för att jag var så självsäker, detta är något jag måste vara redo för i framtiden eftersom ett tag tänkte jag slänga allt jag skrivit och börja om från början. Det är inget man vill hända.

#### Uppgift 1.c planering:

Nummer

Störst

Näst

For

Nummer

If(nummer>störst)

Störst = nummer

If(nummer > näst && nummer < störst)

Näst = nummer

Så planering på 1.c blev lite speciell, det som du nyss läste var mina kladdningar på papper som jag hade. Saken är den att jag var ganska idé lös om hur uppgiften skulle göra, då jag sen fick guidelinjer av Jonas och Damian, de hjälpte till med att ge idén att man bara ska ha 3 variabler, en för numret de skriver in, en för det högsta numret och en för det näst högsta numret därefter man har en massa if-satser de går igenom för att sätta näst högst och högst deras rätta värden. I sin tur så blev det mkt testande på plats och skrivande på tavlan, tyvärr så antecknade jag inte detta. Men i sin tur så jag se detta som ett planerings sätt för små delar av en uppgift, låt oss säga att detta var något man skulle ha i en stor applikation, dvs. man smalas med gruppen man då jobbar med och alla jobbar fram en lösning och tänker ut hur det ska lösas, eftersom det inte är så mycket kod utan mest bara mer avancerad så att rätt if-satser händer vid rätt tillfälle osv.

Det blev mycket tänkande bakom uppgiften så jag ser detta som en ärlig planering. Samt så glömde jag ange en tidmängd som jag skulle anta uppgiften skulle ta.

**Slutvärdering** – I slutändan så blev jag klar med uppgiften, det tog sin tid och man såg fel efter fel efter fel, och de felen fick man sen ändra i hela koden. Jag var inte jätte glad i denna uppgiften, jag är mycket väl medveten om att vi programmerar objektivt men att lösa denna var mer bara irriterande om jag ska ge min ärliga åsikt. Som jag sa så var det problem som bara dök upp för man inte tänkte på de, man kände sig klar sen så får man höra att den fungerar inte om man skriver in tal mindre än 0, därefter så krabbar den om man skriver in 0, osv. osv. det drev en galen.

Så tiden det tog blev inte så lätt att hålla koll på eftersom det blev mycket gå fram och tillbaka i uppgiften men jag skulle gissa på att det tog runt 4 timmar totalt.