

TDDE54/TDDD87/TDIU08/725G92/9AMA73: Tips för Ada.O1 ...

Torbjörn Jonsson <torbjorn.jonsson@liu.se>

Thu 16/09/2021 07:45

To: TDDD87_2021HT_FP <tddd87_2021ht_fp@student.liu.se>; TDDE54_2021HT_AU <tdde54_2021ht_au@student.liu.se>; TDIU08_2021HT_C9 <tdiu08_2021ht_c9@student.liu.se>; 725G92_2021HT_Z3 <725g92_2021ht_z3@student.liu.se>

Hejsan igen.

Denna uppgift är nog den som har mest problemlösning i sig här i början av kursen. Den innehåller de saker som är i princip ALLT man behöver för att senare lösa alla lösbara problem. Jag gissar att ni går med på att det finns personer som bevisat detta och att det som behövs är:

- Sekvens (ett antal satser som görs efter varandra).
- Villkor ("if" [och "case" som vi inte berör så mycket i denna kurs, men ...])
- Upprepning ("for", "while" och "loop") - Vi kommer till en variant av detta i en senare uppgift, men mer om det då.

Detta gör att denna uppgift är lite mer krånglig att ta sig igenom då det finns mycket som går att göra på olika sätt och det är gjort på ett sätt i uppgiften att det kanske ser lätt ut från början, men det finns mycket att tänka på och många saker man behöver prata med varandra och assistenterna om. Fortfarande förstås för att få fram VARFÖR? och HUR? och inte VAD? D.v.s. det är vägen till målet som är det som ger problemlösningstänket.

I denna uppgift är det dessutom så att vi är mer inne på hur man löser uppgiften så det blir antagligen även kompletteringar på hur man löst uppgiften. Givetvis skall det finnas tips om saker som är bra och vad som är gålet också.

Automaträttningen är i denna uppgift inte vattentät. Det går att komma igenom denna utan att ha gjort "rätt". Det gör att ni kanske får tänka om efter att ni "löst" uppgiften. Det är alltså inte bara ok att det "fungerar".

Nu till det som är tips i denna uppgift för att ni skall ha lite mer kött på benen även i denna del.

Vilken upprepning skall jag använda?

Ja det finns några enkla frågor som man bör ställa sig varje gång man kommer till att man skall upprepa i sitt program. De finns i videon och ni får titta där och i boken.

Att välja rätt loop gör att man slipper stora problem senare. Finns många skäl till att inte bara ta den man tycker man kan just nu eller en som man kanske "alltid har använt". Går inte in på detta här då det är ett stort område och mailet blir nog tillräckligt långt ändå.

Tänk bara på att det inte bara är att välja rätt loop. Man skall tänka på hur man avslutar den också. Ett vanligt fel är att man inte tänker på att "<" och ">" faktiskt inte är heltäckande gällande tester.

Vilket sätt är bäst?

Här är det mycket att svara på. Jag gör det enkelt för mig och säger att det antagligen finns 10 sätt att lösa ett problem (ofta fler). Några av dem är antagligen riktigt "dåliga" och ger "fel svar" och andra är "dåliga", men ger "rätt svar" och en del är "ok" och vissa är "super" (kan vara flera).

Att komma fram till den optimala lösningen är oftast ointressant, men att komma ifrån de "dåliga" är åtminstone målet man bör ha. I denna kurs är vi ute efter att ge er bilder av vad som är bra och mindre bra, men inte att ni skall finnas det optimala.

Vad fastnar man på i automaträttningen?

Här finns det en del kul saker som man kommer att råka ut för. Till att börja med samma saker som i Ada.00, men dessa antar vi att ni är med på. Det som tillkommer är att det nu är mer komplext problem så det finns fler testfall som ställer till det.

Ofta är det så att tabellens rader inte har rätt format. Det justerar du lätt när du råkar få ett extra blanktecken eller så. Dock bör ni INTE försöka justera bredder med separata blanktecken (alltså med "Put(' ');" eller liknande beroende på hur stora tal man har att skriva ut. Detta leder till något vi kallar "fullständig uppräknig" som alltså i princip är duplicering av kod som gör (i princip) samma sak. Går att lösa mycket bättre i nästan alla lägen. [Denna del kan man bli kompletterad på i assistenträttningen om den inte tas av automaträttningen förstås.]

Har man t.ex. inte gjort felhanteringen rätt i inmatningen kommer det att bli galet. Man kan till och med hamna i att programmet kraschar (felmeddelandet kommer då under "Felmeddelanden (ERROR)" i automatsvaret. Oftast med "END_ERROR" då de indata som vi har till ditt program inte räcker till (även om det borde göra det).

Om man inte får rätt antal rader i utskriften. Här finns det saker som ställer till det som man inte tänker på från början, men som är en realitet som ni kommer att behöva ha med er.

Beräkningsfel förstås, men det är ju inget konstigt.

OBS! Vi har inte gjort fullständiga tester då det skulle ställa till det helt och ni skulle inte komma igenom uppgiften. Det är till och med så att det är tveksamt om någon överhuvudtaget skulle klara uppgiften då. Inte ens assistenterna om man "brassar" på med de riktigt häftiga fallen. [En utmaning för dem som känner att de vill tävla. :-)]

Det kommer alltså att finnas fall där automaträttningen går till "Godkänt" där assistenterna sen får ta det intressanta i rättningen på deras sida.

Vanliga saker vi ger komplettering på (i assistenträttningen)

1. Att man använder "fel loop". Det vanligaste felet i denna uppgift. Väldigt vanligt med kommentarer av dem som "löst" uppgiften är "det fungerar ju". I denna uppgift är det till viss del sant, men till viss del inte sant. Ni har säkert fått en del att diskutera kring där redan.

En sak som man kan fundera på är t.ex. i inmatningen av första priset. Detta är ETT flyttal som skall matas in. Hur många "Get" behövs för att mata in ETT flyttal? Om du behöver fler har du

antagligen gjort gålet. Antagligen har du följt en modell som gör att du i ett större perspektiv får STORA problem med dubbling av kod och strul beroende på att du rättar saker på ett ställe och missar andra ställen. Vi vill påvisa detta tidigt och kommer under kursen att var noga med att duplicering av kod inte är en bra modell.

Att man behöver sätta en variabel till ett "felaktigt värde" för att t.ex. komma igång med en loop innebär att du antagligen tänkt "bakvänt". Varför skall jag behöva göra fel för att komma igång.

Det skall dock tilläggas att det finns tillfällena när allt ovan ställs på sin spets och det är "fel" att göra "rätt", men inte i denna kurs. :-)

2. Att man "bygger" en "while" av en "loop" och en "if".

3. Att det blir fel antal rader i tabellen. Det mesta av detta tas i automaträttningen, men det finns ibland delar som är gjorda på lite underliga sätt.

Tips

Denna uppgift kommer nog att vara en uppgift där du behöver skicka in flera gånger och där det kan vara bra att fixa småsaker efter det att man fått tillbaka från automaträttningen. Det testas med fler fall än de vi givit som exempel. Detta handlar om att man som programmerare behöver "tänka defensivt". Ibland lite elakt sagt att man ser "användaren som dum" och räknar med att den "alltid gör fel". Förstås inte meningen att var negativ utan tvärtom igen: Vi vill att ni skall hitta en väg där ni får fram bra testfall för att kunna leverera senare.

Det finns också lite saker som är bra att ha som assistenterna ger er (t.ex. hur man avrundar tal eller vad flyttal ställer till med etc.) så prata med dem när du får tillbaka saker du inte riktigt förstår.

Hoppas att detta också ger en liten push framåt och att ni nu fixar Ada.01 galant.

M.v.h.

/TJ

--

//_/_/_/_/_/_/_ Torbjörn Jonsson
/ 013-28 24 67
/ Torbjorn.Jonsson@LiU.SE
//_ IDA/SaS/UPP
//_/_ Institutionen för Datavetenskap
----- Linköpings universitet