

## TDDE54/TDIU08/725G92/TDDD87/9AMA73: Tips för Ada.00 ...

Torbjörn Jonsson <torbjorn.jonsson@liu.se>

Thu 16/09/2021 06:59

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.

Som utlovat alldeles nyss så kommer här ett mail med tips om det som har med Ada.00 att göra. Bara för att komma fortare framåt alltså.

En del text att läsa, men jag hoppas att det är bra saker för er.

### Bufferhantering

Man kan säga att det absolut största problemet med Ada.00 är att komma på hur de olika "Put" som finns fungerar och hur de olika "Get" som finns beter sig. Det är egentligen 90% av denna uppgift. Det handlar om "bufferhantering" både gällande inmatningen ("Get", "Get\_Line" och "Skip\_Line") och utmaningen ("Put", "Put\_Line" och "New\_Line") och att man håller isär dessa två "buffertar".

Vad det gäller buffertarna ("Standard\_Input" från tangentbordet och "Standard\_Output" till skärmen) är de helt separata. "Standard\_Input" påverkas endast av de tre "kommandona" som har med inläsning att göra och likadant för den andra. Mer om detta har ni i FÖ0 och givetvis i boken. Bufferhanteringen är i princip det viktigaste i denna uppgift gällande "problemlösningssdelen" (inte så stor del i denna uppgift, men det finns inslag). Det första problemet är hur man med hjälp av endast en heltalsvariabel kan läsa in 5 heltal och skriva ut dem igen utan att tappa något av de inmatade talen. Många fastnar på denna (även de som har programmerat tidigare) för att man låser sig att man inte har tillräckligt många variabler, men som ni antagligen kommit på så behöver man ju inte mer än en i just detta fall då inmatningsbufferten behåller värdena tills man hämtar dem.

OBS! Bufferhantering finns både för in- och utmatning av data till/från programmet, men det är oftast inmatningen som är den som är "struligast i början".

OBS! En sak som kommer senare, men som jag tar här är att om man skall felsöka sitt program och gör detta genom att skriva ut små meddelanden ("Put" i programmet) för att se var man passerar (inte dumt att göra om det går fel någonstans) behövs att man har koll på vad som händer när ett program "kraschar".

Ett program som kraschar drar med sig "allt" som har med programmet att göra. Även de buffertar som används. Om man då har data kvar i "Standard\_Output"-bufferten är det ett litet problem som uppstår. Det som finns där skriv inte ut på skärmen!

Vad gör detta då? Jo, det ställer till det på det viset att de sista utskriften man gjort inte syns och man vet därför inte var programmet kraschade i värsta fall.

Vad göra i detta fall? Det bästa man kan göra i detta läge är att skriva ut sina "spårutskrifter" (de

man har för att veta hur långt man kommit i programmet) med "Put\_Line" istället för "Put". Put\_Line gör nämligen så att det "begär" att utskriften skall skrivas ut och inte ligga kvar i bufferten (man brukar kalla detta för att man "flushar" bufferten). Det är inte helt säkert att programmet hinner göra detta, men det är bra mycket säkrare och det ger er bättre chans att finna era fel senare.

Hett tips alltså: Kolla på detta om hur man gör en enkel felsökning i sina program. Prat med assistenten om det också. Finns lite om det på hemsidan (vi pratar enkel felsökning i denna kurs och vi går inte igenom de mer avancerade metoderna då det ligger i fortsättningskurserna).

## Exakta utskrifter

Att få till exakt rätt utskrifter är det som jag tror att de flesta av er sitter och har lite "problem med" innan ni kommer igenom automaträttningen. Här handlar det mest om att vara noggrann gällande vad man skall skriva, men det struligaste i början brukar vara att komma på vilken "Put" man skall använda och hur dessa betar sig. T.ex. skriver "Put" för "Character" och "String" ut exakt det som står inom parenteserna, medan "Put" för "Integer" och "Float" i standardläget (utan en massa extra formatdirektiv skriver ut talen på lite speciella sätt.

Tag t.ex. utskriften av heltal. Skriver man "Put(l);" (där variabeln "l" är en "Integer") så skrivs heltalet i variabeln ut, men med en massa extra blanktecken framför talet. Varför det??? Jo, det finns en standardinställning på "formatet" på utskriften som är att talet skall skrivas ut med 11 teckens bredd. VA??? Varför då? Jo, ett heltal kan anta värden som ligger i ungefär intervallet -2\_000\_000\_000 .. 2\_000\_000\_000 (lite drygt i båda riktningarna) och för att inte det skall vara strul med att talet inte skall få plats valde man att sätta 11 tecken som utrymme då det inkluderar även minustecknet på den negativa sidan). Motsvarande för flyttal är att talet skrivs ut i "exponentialform" (t.ex. skulle "1.23456E2" var ett exempel på hur talet "123.456" skulle kunna skrivas ut).

Hur man styr dessa utskrifter har ni säkert koll på nu. Att "Width" gör att man kan styra hur heltal skall skrivas ut och "Fore", "Aft" och "Exp" styr hur flyttalet skrivs ut. Dessa formatteringsdelar till "Put" för heltal och flyttal finns inte för strängar och tecken som ni också har sett.

Det som är ett problem som oftast dyker upp i alla uppgifter är att man skall skriva ut tabeller eller saker på visst format. Det är därför vi har denna Ada.00 för att ni skall ha koll på alla dessa saker.

Utskriftsformaten är nog det som ställer till det mest för er när ni skall ta er igenom automaträttningen om man ser till antalet "fel" man kan göra. Det är många kombinationer av saker som ställer till det och det är därför vi har inte haft någon brådska att trycka på att ni skall vara klara med Ada.00 tidigare. Nu börjar det dock bli dags att rulla igenom denna så här kommer de hetaste tipsen om vad som går fel i "automaträttningen" så att ni kan "undvika dem".

1. Felaktiga "Width" respektive "Fore", "Aft", "Exp" gör att det fattas eller blir för många blanktecken före de tal som skrivs ut. Här är det skillnad på att skriva ut ETT blanktecken mellan varje data och att få till en generell mängd tecken för heltalet (eller heltalsdelen i ett flyttal). I vissa delar i uppgiften måste man kolla de olika körexemplen för att se om man skall skriva ut en "Character" (ett blanktecken) eller om det är formattering av tal som gäller.

2. Felaktig text i utskrifterna (t.ex. "Du skrev" istället för "Du matade in" eller kanske ännu mindre saker). Dessa är ju triviala att rätta till så om du får tillbaka något från automaträttningen med sådana fel så fixa det och skicka in igen så fort som möjligt så att du inte sitter med det flera dagar och väntar.

3. Extra blanktecken i slutet av rader av utskrifter. Dessa är besvärliga att se. Automaträttningen är "petig" här och säger att det är fel, men man ser ju inte blanktecken i det meddelande man får då det ju är "genomskinligt". Kolla ert program noga så att ni inte skriver ut extra blanktecken i slutet av raderna är ett hett tips.

I princip hela denna uppgift handlar om de "kommandon" som står längre upp i mailet ("Get", "Put", ...). Det finns en liten sak till som också finns med och det är att när man använder "Get\_Line". "Get\_Line" betar sig på två olika sätt beroende på hur lång sträng användaren matar in (d.v.s. det har med bufferhanteringen att göra). Detta gör att man behöver två "if"-satser i slutet av uppgiften för att säkerställa att allt blir rätt (man kanske tycker att det bara behövs en "if", men det är bra att göra lika på den sista inmatningen också). Mer om detta finns dels i boken och det finns också en del extra om detta bland det extra material som finns vid FÖ-filmerna.

### **Teckenkodning (teckenuppsättningar)**

I dagens datorer används olika teckenuppsättningar överallt (olika benämningar på dessa kan vara UTF-8, Latin-1, ISO-8859-1, ...). Det kan vara skillnader mellan Windows, Mac och Linux. D.v.s. att man kör på olika datorer med olika operativsystem, men det kan lika gärna vara att man har ställt om sin terminal att använda en annan teckenkodning eller att man använder olika typer av program (Emacs, Word, ...).

I denna kurs råkar ni antagligen mest ut för att man sitter på olika operativsystem. Det finns t.ex. skillnader även inom en teckenkodning som t.ex. UTF-8 som gör att det skiljer sig mellan Windows och Linux. En besvärlig situation helt enkelt och detta är bara som det är tyvärr.

Ni har om ni skickat in saker till automaträttningen sett effekter av detta (vi arbetar på att få bort de, men prioriterar en del annat högre just nu bara) i detta med att "ääö" ser "underliga ut". Det spelar inte så stor roll så länge vi har samma teckenkodning i vårt program som det ni skickar in (och vi försöker detektera vad ni råkar ha för teckenkodning och byter denna till "vår variant" om det går).

Det som fortfarande är lite struligt är att det smyger in lite udda tecken på olika ställen. Vi har nu sett att även "-tecknet är ett besvärligt tecken. Det finns flera OLIKA sådana som alltså har olika "kod" även inom samma teckenkodning. Detta gör att just detta tecken inte går att klistra in från PDF:en direkt som inmatning i programmet. Det är ett annat " än det man får om man trycker på tangentbordet i terminalen. :( Det är skapat från "word" som tydligen tyckte att det var smidigt att använda den varianten av just citationstecknet.

Effekterna gör dock att ni behöver ett par extra tips.

1. Klistra inte in saker från PDF:erna (eller var beredd på att det kan ge ett extra steg i automaträttningen). Om ni råkat göra detta är det förstås inte mer än att ni skickar in er kod igen med tillfixat tecken.

2. I detta kommer förstås också "åööüü..." (en massa tecken som har olika positioner i olika teckentabeller). Vi räknar med att ni kör med standardinställningarna i ThinLinc och på skolans datorer och då bör ni ha UTF-8 som teckenkodning och allt blir då lika som våra program ger som resultat (även om det kan se lite strul ut med dessa tecken i de returmeddelanden ni får är det iallafall samma och då är det inget problem).

3. Om det är så att ni trots att det verkar vara rätt tecken (t.ex. "åöö") får en differens i automaträttningsmailet kolla med assistenten om det är ett annat tecken det skall vara. Vi får tips från dem om det är så att något är galet.

## Avslutning

Jag hoppas nu att ni kommer att rulla in med lösningar på denna uppgift och att ni inte ger upp bara för att det kommer en del automaträttningsvar som gör att man behöver fixa till lite och skicka in igen.

Tag chansen att skicka in igen samma dag (flera gånger om så behövs) för att komma framåt.

Ha nu en trevlig dag och lycka till med programmerandet.

M.v.h.

/TJ

--

-----  
\_/\_/\_/\_/\_/\_/\_/\_/ **Torbjörn Jonsson**  
\_/\_/           \_/\_/ **013-28 24 67**  
\_/\_/           \_/\_/ [Torbjorn.Jonsson@LiU.SE](mailto:Torbjorn.Jonsson@LiU.SE)  
\_/\_/   \_/\_/   \_/\_/ **IDA/SaS/UPP**  
\_/\_/   \_/\_/\_/ **Institutionen för Datavetenskap**  
----- **Linköpings universitet**