

Programeringsstuga 1

Programspecifikation - Temperaturtabell

Programmet skall fråga användaren efter en starttemperatur, en sluttemperatur och en steglängd. Därefter skall programmet skriva ut en tabell i tre kolumner. Mittenkolumnen skall börja på angivet startvärde och i steg om angiven steglängd sluta så nära exakta slutvärdet som möjligt. Värden efter slutvärdet skall inte komma med i tabellen. Mittenkolumnen representerar en temperatur i antingen Celsius eller Fahrenheit. Vänsterkolumnen tolkar mittenkolumnen som grader Fahrenheit och skriver ut detta värde i Celsius. Högerkolumnen tolkar mittenkolumnen som grader Celsius och skriver ut detta värde i Fahrenheit. Användaren förutsätts mata in giltiga flyttal med maximalt 2 deciamler. Programmet skall göra den felkontroll som behövs för att programmet skall fungera korrekt givet indata enligt förutsättningarna. Programmet skall även kontrollera att inmatat intervall är rimligt för temperaturer i antingen Celsius eller Fahrenheit. Indata utanför förutsättningarna hanteras inte. Programmet skall ha stöd för tabeller i både stigande och fallande skala beroende på indata. Alla beräkningar skall ske så att två decimalers noggrannhet bibehålls.

Förberedelse

Sex programmerare har tagit fram var sin lösning av varierande kvalitet. Varje lösning innehåller något bra och något dåligt. Det kan vara allt från dålig kodformatering till bra algoritm. Studera lösningen du blivit tilldelad så du förstår den. Det går bra att fråga examinator i förväg om det är något du tycker är konstigt. Eller så sparar du frågorna tills programmeringsstugan så ni kan diskutera tillsammans. Tänk igenom fördelar och nackdelar med den lösning du blivit tilldelad. Testkör programmet och se hur det fungerar. Uppfyller det specifikationen? Fundera igenom vad programmeraren gjort för fel och hur det kan åtgärdas.

Du får själv hitta testfall (indata) för att testa att specifikationen är uppfylld. Försök hitta något testfall för varje krav. Här kommer några kommandoradsknep du kan använda, och även ett par testfall:

```
g++ solutionX.cc
echo 0 11 1 | ./a.out
echo 11 0 1 | ./a.out
echo 0 11 .1 | ./a.out
echo 0 110000 .1 | ./a.out | tail
```

Du kan naturligtvis skriva in indata själv när programmet startar, men genom att använda "echo" som skickar in indata till programmet blir det lättare att testa igen. Anledningen att "tail" används är att utskrift till skärm är mycket långsamt, så genom att bara ta med de sista raderna går programkörningen mycket snabbare när stora tabeller skall testas. Du rekommenderas ändå att inte testa med större indata än 200000.

Förberedelsen beräknas ta 2 - 4 timmar i anspråk. Du bör vara så insatt att du själv snabbt kan skriva en bra lösning på problemet.

Uppgift på dojo

På programmeringsstugan skall du (informellt) presentera den lösning du blivit tilldelad, hur den fungerar, dess fördelar och dess nackdelar. Du bör ha hittat något bra, något dåligt och något du vill fråga om. Några exempel på saker som kan vara bra att granska:

- Namngivning
- Kodstil
- Kommentarer
- Val av loop
- Val av lösning
- Korrekthet

Presentationen får ta den tid den tar, men cirka 5 minuter är kanske lämpligt för att scrolla igenom hela lösningen och lyfta fram det som verkar speciellt, samt visa några intressanta testfall.

Lösningarna skiljer sig åt en del, och det ingår att ni tillsammans skall reda ut varför det är gjort som det är gjort om frågetecken finns. En lösning som först verkar bra/dålig kan tänkas komma i annan dager när någon annan presenterat sin tilldelade lösning.

Publiken måste under "presentationen" vara observant på skillnader och likheter i deras tilldelade lösning, samt ställa frågor om varje lösning (assistent kommer närvara hela tiden om ni behöver input utanför gruppen).

När alla presenterat sin tilldelade lösning skall gruppen tillsammans (6 personer) först diskutera hur en så bra lösning som möjligt skall se ut, och sedan skriva en sådan tillsammans. Det går bra att utgå från en av de givna "lösningarna".