

Välkomna alla SUVx-23-are till **Boiler room fre 20/10, kl 9-12**. Här kommer mina två uppdrag till klassen för dagens Boiler room.

1. Uppdrag Bergvärme

En borrhutrustning för bergvärme kan borra 25 m under den 1:a timmen i en viss tomtmark.

Under de följande timmarna minskar borrens prestation med uppskattningsvis 10-20% per timme. Den exakta minskningen är inte känd, då den är beroende av markförhållandena. Borren ska gå oavbrutet i 8 timmar.

Skriv ett program som uppskattar det totala borrhjupet.

Börja med att simulera minskningen av borrens prestation efter den 1:a timmen med slumpstal mellan 10 och 20. Summera borrhålets djup efter den 1:a timmen baserad på denna simulation.

Skriv ut slutligen ett närmevärde för borrhålets totala djup efter 8 timmar.

Skriv ut även borrhjupets procentuella minskning per timme vid den aktuella körningen, t.ex.:

"Detta närmevärde baseras på en 14%-ig minskning av borrens prestationen per timme efter den första timmen."

P.g.a. simuleringen med slumpstal borde man få vid olika körningar olika procentsatser för minskningen av borrens prestation och därmed även andra uppskattningar av det totala borrhjupet. I praktiken duger dock ofta en sådan uppsättning som kan vara en värdefull information för planeringen av arbetet.

Frivilligt! När programmet fungerar fundera på en vidareutveckling av uppdraget som kan lyda så här:

Lös det omvända problemet: Skriv ett program som läser in ett önskat totalt borrhjup och beräknar samt skriver ut antalet arbetstimmar som borren skulle behöva för att nå detta borrhjup.

2. Uppdrag Kalkylator

I denna uppgift ska ni skapa ett program **Calculator** som stödjer följande funktionaliteter: addition, subtraktion, multiplikation, division och potentiering samt att kunna ange det största och minsta av två inmatade tal.

Kalkylatorn ska vara igång kontinuerligt tills användaren väljer att stänga av den. Detta kan göras genom att bygga in en loop. De olika räkneoperationerna ska definieras i separata funktioner och anropas i **main()**.

Följande funktioner ska definieras i programmet **Calculator**:

```
double add(double operand1, double operand2)
{
    // Addition av operand1 och operand2
}

double sub(double operand1, double operand2)
{
    // Subtraktion: operand1 - operand2
    // Även subtraktion av negativa tal
}

double mult(double operand1, double operand2)
{
    // Multiplikation av operand1 med operand2
}

double div(double operand1, double operand2)
{
    // operand1 / operand2
    // Division med 0 ska förhindras
    // Felmeddelande vid inmatning av 0 för operand2
}
```

```
double pow(double operand1, double operand2)
{
    // Beräkning av potens: operand1 upphöjt till operand2
}

double max(double operand1, double operand2)
{
    // Returnera det större värdet av operand1 och operand2
}

double min(double operand1, double operand2)
{
    // Returnera det mindre värdet av operand1 och operand2
}
```

Programmet skall exekvera kontinuerligt tills användaren väljer att avsluta körningen. För att åstadkomma detta kan man använda t.ex. en `do-sats`. Kalkylatorn kan avslutas genom att användaren matar in t.ex. tecknet `q` (quit) istället för en operator.

Placera först, när ni börjar koda, all kod i en fil. Flytta sedan alla ovannämnda funktioner till en annan fil. Bibehåll endast `main()` som läser in data, anropar funktionerna och skriver ut resultat.

Frivilligt! I slutet, när allt fungerar, försök att lägga in kod som hanterar ev. felaktiga inmatningar. Detta borde inkludera både inmatning av operander, men även av operatorer.

Se till att ni skriver ut meningsfulla felmeddelanden, så att användaren får möjligheten att rätta till sin felaktiga inmatning.

Lycka till!

Hälsningar

Taifun