

Programmering 1

Laboration 4: Mera beslutsfattande och iteration

Denna laboration är tänkt att både utföras och **redovisas** under de schemalagda timmar som finns anslagna för ändamålet. Ifall du inte har möjlighet att delta i laborationstillfällena, eller inte hinner slutföra laborationen under den schemalagda tiden måste du redovisa dina resultat i en **skriftlig laborationsrapport** som skickas in via Moodle. Ett obligatoriskt delkrav för att kunna bli godkänd i kursen är att **alla laborationer är godkända**.

Obs: Vissa uppgifter (markerade med en asterisk) kommer att gås igenom gemensamt under laborationstillfällena. Dessa uppgifter godkänns alltså automatiskt om du deltar i laborationstillfället.

1. (Något mera avancerad) talsortering

Skriv ett program som läser in tre heltal från användaren, sorterar dem och skriver ut dem från det minsta till det största talet. Använd dig av **logiska operatorer** och **kombinerade logiska uttryck** för att utföra sorteringen.

En programkörning kan t.ex. se ut som nedan:

```
Please input three values: 2 4 1
```

```
The sorted list is:
```

```
1
2
4
```

Kommentar: Sortering är en klassisk uppgift för en dator. I kursen *Programmering 2* kommer du att få bekanta dig med effektiva algoritmer för att sortera stora mängder element.

2. Utökad betygskalkylator med inputkontroll

Utöka din betygskalkylator från föregående laboration så att den kontrollerar om de inmatade värdena håller sig inom de givna gränsvärdena. Ifall något annat värde matas in skall frågan upprepas. Använd dig av kombinerade logiska uttryck för att implementera kontrollen.

```
Please enter the exam result (0-30 points): 35
Please enter the exam result (0-30 points): 25
Please enter the assignment #1 result (0-10 points): 6
Please enter the assignment #2 result (0-10 points): 10
Please enter the assignment #3 result (0-10 points): 9
Please enter the assignment #4 result (0-10 points): 10
Please enter the assignment #5 result (0-10 points): 12
Please enter the assignment #5 result (0-10 points): 8
```

```
The final grade is 84.7%, which corresponds to the VG grade
```

3. *For-loop, upprepade potensberäkningar*

Skriv ett program (dvs. modifiera uppgift 3 från föregående laboration) som med hjälp av en **for**-loop skriver ut följande resultat:

```
Iteration 1: 2^1 is 2
Iteration 2: 2^2 is 4
Iteration 3: 2^3 is 8
Iteration 4: 2^4 is 16
Iteration 5: 2^5 is 32
```

4. *Omsvängd for-loop*

Skriv ett program som med hjälp av en for-loop skriver ut följande resultat:

```
Iteration 5: 2^5 is 32!
Iteration 4: 2^4 is 16!
Iteration 3: 2^3 is 8!
Iteration 2: 2^2 is 4!
Iteration 1: 2^1 is 2!
```

5. (*) *Nästlad for-loop 1*

Skriv ett program som ritar upp en fylld rektangel med stjärnor (*) utgående från inläst bredd och höjd. Använd dig av nästlade **for**-loopar.

En programkörning kan se ut som nedan:

```
Enter width and height: 8 4
*****
*****
*****
*****
```

6. (*) *Nästlad for-loop 2*

Modifiera din lösning på uppgift 6 så att det istället skriver ut en ram med de angivna dimensionerna.

En programkörning kan se ut som nedan:

```
Enter width and height: 8 4
*****
*          *
*          *
*****
```

7. (*) Kalkylator med switch

Implementera en heltalskalkylator för addition/subtraktion/multiplikation/division/modulus. Använd en **switch**-sats för att välja räknesätt.

Enter an expression: 13+5 <== scanf("%d%c%d", ...)
Result = 18

Enter an expression: 13/5
Result = 2

Enter an expression: 13%5
Result = 3 <== "3 femtedelar"

Enter an expression: 13^5
Unknown operator: ^