

Laboration 2

Syfte:

- Lära dig att läsa in data från tangentbordet till variabler, att utföra beräkningssatser och att presentera resultat på bildskärmen.

Redovisning:

- Skriv en sammanfattning av ditt arbete. Lämna in på IT's i pdf-format. Du kan lämna in individuellt eller tillsammans med (max) en kamrat.
Sammanfattningen ska innehålla punkterna:
 - ❖ Programkoden för den uppgift av nedanstående du är mest stolt över.
 - ❖ Genomförande (vad var lätt, vad var svårt).
 - ❖ Resultat (vad du lärde dig).
 - ❖ **Tips:**
 - Dokumentera ditt arbete under tiden du tar dig igenom anvisningarna.

Uppgifter

1. Skriv ett program som ber användaren skriva in sitt namn. Programmet skriver sedan ut "Hej Kalle!" (om man nu skrev Kalle).
2. Skriv ett program som frågar efter radien i en cirkel och sedan skriver ut vad arean är för en sådan cirkel. Deklarera pi som en flyttalskonstant (inte med DEFINE).
3. Skriv ett program som läser in namnet på två "spelare" vilka därefter ska göra två tärningskast var. Slutligen ska programmet presentera summan av tärningarnas värde för respektive spelare. Nedanstående ruta visar ett exempel på en körning av en lösning på programmet, alla fetstilta uppgifter är inmatade av användaren.

```
Spelare 1: Kalle Stropp
Spelare 2: Grodan Boll

Tärningskast för Kalle Stropp : 3 och 4
Tärningskast för Grodan Boll : 6 och 2

Kalle Stropp fick resultatet 7
Grodan Boll fick resultatet 8
```

Tips!

I biblioteket stdlib finns funktioner för att generera s.k. pseudoslumptal. Pseudo betyder falsk, numren är bara skenbart slumpmässiga, men de duger för våra syften. **Inkludera stdlib.h** i ditt program när du vill använda slumptal. Använd funktionen rand() för att få ett slumptal. Gör så här:

```
int aRandomNumber;  
/* Talet efter % tecknet anger spännvidden på de tal man vill ha.  
   Talet efter + tecknet anger det minsta talet man vill ha.*/  
aRandomNumber = rand() % 6 + 1;  
/*Här vill vi kunna få 6 olika slumpstal, det minsta ska vara 1.*/
```

Varje gång man kör programmet kommer samma "slumptal" upp, de hämtas bara från en lång lista av tal, samma varje gång. För att få en annan lista med "slumptal" ger man den s.k. slumpstalsgeneratoren ett "frö" (eng. seed) att bygga talserien från. För att få olika frö varje gång man kör programmet, använder man datorns klocka: `time(NULL)` ger aktuell tid. (I stället för `NULL` kan man ange att man t.ex. vill ha tiden i Australien). Detta klockslag ger vi sedan till slumpstalsgeneratoren på följande sätt: **`srand(time(NULL));`**
Lägga alltså in den raden innan du börjar använda `rand()`. OBS: `srand` ska bara göras **en** gång. För att kunna använda `time` måste du **inkludera `<time.h>`**.

4. Skriv ett program som frågar efter två sidor i en rätvinklig triangel och sedan skriver ut hypotenusans längd (Pythagoras sats). (Jämför uppgift 6 lab 1)

5.

Skriv ett program som konstruerar inbjudningskort (på bildskärmen). Användaren ska mata in namnet på den som gör inbjudan, datum, bjudningstid, hur många timmar festen varar, plats och namnet på den som ska bjudas och därefter ska inbjudningskortet genereras.

Nedanstående ruta visar ett exempel på en körning av en lösning på programmet, alla fetstilta uppgifter är inmatade av användaren.

```
Vem står för inbjudan? Britta  
Vilket datum? 27 maj  
Vilket klockslag börjar festen? 18  
Hur många timmar håller festen på? 4  
Var? Klubbstugan  
Vem ska bjudas? Anna  
  
-----  
Hej Anna!  
  
Du är välkommen på fest den 27 maj kl 18 i Klubbstugan. Festen varar  
till kockan 22.  
Hälsningar  
Britta
```

6. Låt ett program läsa in ett heltal och sedan skriva ut det i hexadecimal, oktal och binär form.
7. Konstruera ett program som frågar efter 4 material och deras specifika vikt. Programmet presenterar sedan de inlästa uppgifterna i en snyggt formaterad tabell enligt modell från föreläsningen (alla materialnamnen under varandra och vänsterjusterade, värdena skrivs ut högerjusterade och med 3 decimaler).