

Exercise 4

Du burde på nuværende tidspunkt være fortrolig med, hvordan og hvornår du skal oprette flere projekter i samme solution i Visual Studio. Dette vil derfor ikke være beskrevet i opgaverne fremover – nu skal du selv ”styre” det 😊

Exercise 4.1 En simpel funktion

- Skriv prototypen til en funktion `sumBiggerThan100()`, som kan modtage to decimaltal og returnere et helt tal (HINT 4A).
NB! Husk, at du skal skrive prototypen i en header-fil (*.h).
- Funktionen skal returnere 1 (true), hvis summen af de to tal er større end 100, og 0 (false) hvis summen er mindre end eller lig med 100. Skriv nu koden (implementering) til funktionen.
NB! Husk, at du skal skrive implementeringen i en selvstændig source-fil (*.cpp).
Skriv pseudokode for *funktionen* inden du koder den.
- Skriv et lille testprogram (`main()`) til din funktion (ny source-fil).

NB! Fra nu af og fremover skriver du selv pseudokode til opgaverne, selvom du ikke bliver bedt direkte om det i opgaven 😊

Exercise 4.2 Trafikstyring – forbedret programstruktur

- Skriv et modul (header- og sourcefil) med **to** funktioner. Den ene skal hedde `redToGreen()` og skal få ”trafikstanderen” til at **skifte** fra rød til grøn. Den anden skal hedde `greenToRed()` og skal få trafik-standeren til at **skifte** fra grøn til rød.
- Brug herefter disse to funktioner i din løsning til exercise 3.4 (eller 3.5) fra sidste uge.
NB! Læg mærke til hvor meget det simplificerer din main-funktion – hvilket jo betyder, at din kode bliver mere struktureret 😊

Fortsættes næste side

Exercise 4.3

Skriv et modul (header- og sourcefil) med følgende to funktioner:

```
int isEven(int number);  
int isInteger(double number);
```

Den første funktion skal returnere 1 (true), hvis number er et lige tal ellers returneres 0 (false).

Den anden funktion skal returnere 1 (true) hvis number er et helt tal (dvs. at decimalerne er 0.00000) ellers returneres 0 (false).

Skriv herefter et program (main), hvor i du tester dine funktioner.

Exercise 4.4 Blinkfunktioner - RPi

Skriv et modul (header- og sourcefil) med to funktioner som udfører hver sin blinke-sekvens med en lysdiode på RPi'en. Du vælger helt selv hvilke blinke-sekvenser du vil have funktionerne til at udfører, men der skal være tydelig forskel på dem og den ene skal fortsætte i 5 sek. når den aktiveres og den anden i 10 sek.

Ekstraopgaver

Exercise 4.5 Termometer - RPi

Skriv et lille program som viser temperaturen vha. lysdioderne på RPi'en.

Hvis temperauren er større end eller lig med 22 gr. skal der lyse en grøn lysdiode. Hvis temperaturen er større end eller lig med 23 gr. skal der lyse to grønne lysdioder. Osv. Dvs. at hvis temperaturen er større end eller lig med 27 gr. skal alle 6 lysdioder lyse. Temperaturen skal måles og lysdioderne opdateres en gang pr. sek. Programmet skal forsætte indtil der trykkes på en tast på PC tastaturet.

Exercise 4.6 Styring af lysdioder - RPi

Du skal lave et program der tænder/slukker lysdioderne på RPi som reaktion på at brugeren trykker en tast på PC'ens tastatur. Tasterne fra '1' til '6' skal tænde de respektive lysdioder LD1 til LD6. Tasten '0' skal stoppe programmet. I første udgave (version 1.0) af programmet skal man ikke kunne slukke lysdioderne.

Fortsættes næste side

En løsningsmulighed er først at læse input fra tastaturet, indtil en korrekt værdi er indtastet, fx med en `do/while`. (HINT 4B).

Dernæst kan man bruge en `switch` til at udføre den ønskede aktivitet, som beskrevet.

Skriv først en kladde til programmet i form af pseudokode.

Forslag til udvidelser/ændringer:

Version 1.1

Når en lysdiode tændes, skal den forrige, der blev tændt, slukkes.

Version 1.2

Tasterne "toggler" den tilsvarende lysdiode, uden at ændre på de andre. Dvs., når en lysdiode er slukket, og dens tast trykkes, skal den tændes, og omvendt. (HINT 4C).

Version 1.3

I stedet for en `switch`, kan man "regne" sig frem til hvilken lysdiode, der skal tændes eller slukkes. Parameterværdierne `PD1` til `PD6` til `ledOn()` og `ledOff()` dækker jo over tallene 1 til 6. Tegnene '1' til '6' har ASCII værdierne 49 til 54. Kan du skrive noget kode, der omregner fra ASCII værdi til lysdiodenummer? (HINT 4D).

Denne ændring giver ikke så meget mening, hvis lysdioden skal toggles, men prøv ændringen på version 1.0 eller version 1.1.

Hint 4A: Hvis en funktion skal kunne modtage to decimaltal og returnere et helt tal skal dens prototype enten være

```
int sumBiggerThan100( float, float );
```

eller

```
int sumBiggerThan100( double, double );
```

Hint 4B: Brug `_getch()` eller `scanf_s("\n%c", &input)` som du gjorde i opgave 3.5 sidste uge.

Hint 4C: Du får brug for huske, hvad den aktuelle tilstand for hver lysdiode er, så du i de forskelle cases i `switch` sætningen kan tage en beslutning om der skal tændes eller slukkes. Lav fx 6 heltals variable der hedder `on1`, ... `on6`, som indeholder 0 hvis den tilsvarende lysdiode aktuelt er slukket, 1, hvis den er tændt.

Hint 4D: Lavede du Home Exercise 3.3 i sidste uge? Ved at lave den, lærer du noget om heltalsværdier, tegn og ASCII værdier, som du kan bruge her.