

# Torsdags opgave 5

September 29, 2016

Skrevet af: Markus Visvaldis Ingemann Thieden  
Studienummer: s164920

kode udarbejdet sammen med:  
s153460 Jonas Ladefoged Holm  
s113070 David Bjerre Bjørklund

Besvarelsen består af følgende filer:

**Kode:**

uge51.c  
uge52.c  
uge53.c  
uge54.c

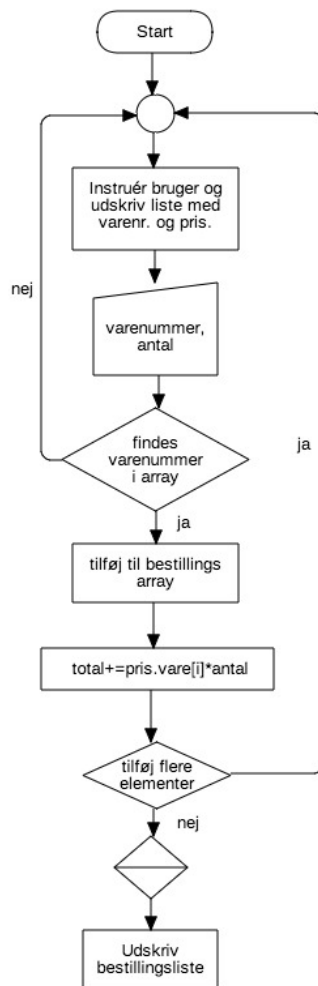
**tekstfiler til opgave 5.2**

inputfil.txt  
forventetoutput.txt

## uge 5.1

Lav et program, hvor man indtaster en kundeordre/varebestilling i form af varenummer og antal. Firmaets lagervarer - varenumre, varebeskrivelse og priser - skal være gemt i arrays. Programmet skal anvende pointers til at udskrive en komplet liste over de bestilte varer samt en samlet pris.  
De relevante filer er: uge51.c, uge5.h

**Flowchart:**



Vi starter med at udskrive en komplet liste over mulige varer, og instruktioner som til at bruge programmet. Derefter skriver brugeren den ønsket vare samt antal, hvis varen ikke findes bliver brugeren bedt om at indtaste igen. Derefter bliver inputet lagt i et bestillings array, hvis bruger har flere ønsket vare bliver det loopet, ellers printes bestillings listen.

### Implementering

Vi bruger void pointere til at printe vores array, vi gemmer værdien i void pointeren, og printer den derfra. For at kunne printe vores void pointer typecaster vi den til den korrekte form for pointer, og derefererer den.

```
void *ordreAllRoundPointer;
```

```

    ordreAllRoundPointer = &ordre[i].vareNr;
    printf("\t\t\t%2d",*(int*)ordreAllRoundPointer);

    ordreAllRoundPointer = &firmaLager[temp].pris;
    printf("%16dkr",*(int*)ordreAllRoundPointer);

    ordreAllRoundPointer = &ordre[i].antal;
    printf("%9d",*(int*)ordreAllRoundPointer);

    ordreAllRoundPointer = &subtotal;
    printf("%12dkr",*(int*)ordreAllRoundPointer);

    ordreAllRoundPointer = &firmaLager[temp].beskrivelse;
    printf(filPointer, "\t%s\n", *(char**)ordreAllRoundPointer);

```

Til sidst typecaster vi som en char\*\*, det gør vi fordi at en streng allerede er defineret som en char\*, da vores void pointer peger på char pointeren

### Testdokumentation

Vores program er ikke særligt robust, hvis man ikke giver de ønsket input i den rigtige datatype kan programmet finde på at terminere med det samme. Eller hvis man giver et forkert slags input når man skriver vare nummer ind bliver man fanget i et uendeligt loop.

## uge 5.2

Dit program fra Uge4,1 skal sortere at den indtastede ordre i stigende varenummer ved brug af pointere.

Programmet skal anvende pointere til at udskrive/aflevere en komplet liste til en fil over de bestilte varer samt en samlet pris.

De relevante filer er: uge52.c, uge5.h, input.txt, forventetoutput.txt

### Implementering

Vores printe funktion er blevet ændret til at åbne en fil op og skrive til den i stedet for at den skriver til standard output. fprintf bruges i stedet for printf.

```

FILE *filPointer;
filPointer = fopen("Bestilling.txt", "w");

fprintf(filPointer, "\n\n\tBestilling:\n");

```

### Testdokumentation

Vi har lavet en input fil så man kan teste programmet med

```
./uge52<input.txt
```

Den outputfilen Bestilling.txt skulle være ens med forventetoutput.txt

ellers

goto **uge 5.1 Testdokumentation;**

## uge 5.3

Gennemarbejd og kommenter programmet i fig. 7.28, der kan anvendes til et menu-system. Programmet afprøves og dokumenteres. De relevante filer er: uge53.c

### Implementering

Vi har copy pasted fra bogen.

Programmet erklærer tre funktioner der bare printer deres input. De funktioner bliver brugt til initialiseringen af en funktion pointer array. Derefter køre et while loop hvor man får lov til at vælge hvilke funktioner der skal køres.

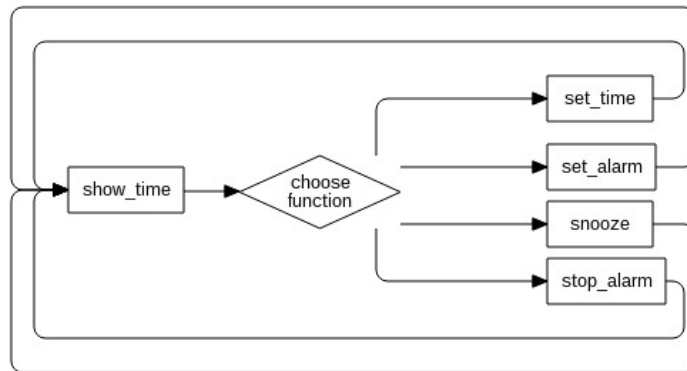
### Testdokumentation

Vi har testet programmet med forskellige input, talene 1-3 fungerer som forventet og får programmet til at eksekvere den pågældende funktion. Alle andre input i form af strenge og lignende får programmet til at slutte.

## uge 5.4

Lav et menu-system (ved brug af pointere til funktioner) til et pc-baseret vækkeur. De relevante filer er: uge54.c

### Flowchart:



Vores program køre i en uendelig løkke, hvor at man giver et input som aktivere en givet funktion, hvorefter at tiden bliver vist.

### Implementering

Vi har brugt time.h til at bestemme tiden og gemme tiden, vi bruger time.h egen datastruktur til at gemme tiden, og vi hiver data ud af dem. De brugte funktioner og data strukturer er dokumenteret i koden

Vores funktioner er nogle simple nogen der bare printer tiden og lignende og ændre værdien i nogle globale variable, kik i koden for at se hvad de forskellige funktioner gør.

Vores i vores main funktion initialisere vi et funktion pointer array med alle vores funktioner.

```
void (*menu[6])(void) = { show_time,
                          set_time,
                          set_alarm,
                          snooze,
                          stop_alarm,
                          power_off};
```

Derefter giver brugeren et input, der aflæser hvilken funktion fra arrayet som brugeren har valgt.

```
scanf(" %d",&choice);

choice--;
(*menu[choice])();
```

### **Testdokumentation**

Vores program virker ikke hvis man giver et input der ikke er et tal, char og strenge fanger vores program brugeren i en uendelig løkke.