

Hardwarenær programmering E16

Uge 3, bestillingssystem med bubblesort og structs

Afleveret 22. September 2016

David Bjerre Bjørklund
s113070

Formål og krav

Ugens opgave bygger på opgaven fra uge 2, derfor vil meget af koden også være lignende. Igen består opgaven af 4 delopgaver som herunder beskrives med krav.

Første delopgave

Programmet skal tage to input, et varenummer og antal. Varenummer skal krydsrefereres med en liste over produkter virksomheden tilbyder. Programmet giver et output i form af en udskrift af bestillingerne, samt en samlet pris.

Anden delopgave

Denne opgave er mest en teknikalitet. Programmet skal tage et inputstrøm i form af en fil og udskrive til anden anden fil.

Tredje delopgave

Indtastede ordrer skal sorteres. Herefter skal det indtastede varenummer ligge i intervallet 0-100. Ydermere må der ikke bestilles mere end 5 stk af hvert produkt. Ved ugyldig indtastning skal der udskrives en fejlmeddelelse

Fjerde delopgave

Programmet håndtere rabat og forsendelse, efter følgende betingelser.

total < 200: +50 kr i forsendelse

200 <= total < 500: fri forsendelse

total >= 500: fri fragt + 10% rabat

Efterfølgende skal der udskrives en "faktura" med de bestilte varer, priser, samt subtotal, total, rabat og forsendelse.

Betingelsen for forsendelse ved færre end 3 varer er ikke en meningsfuld betingelse da denne allerede er dækket ind under betingelse 1.

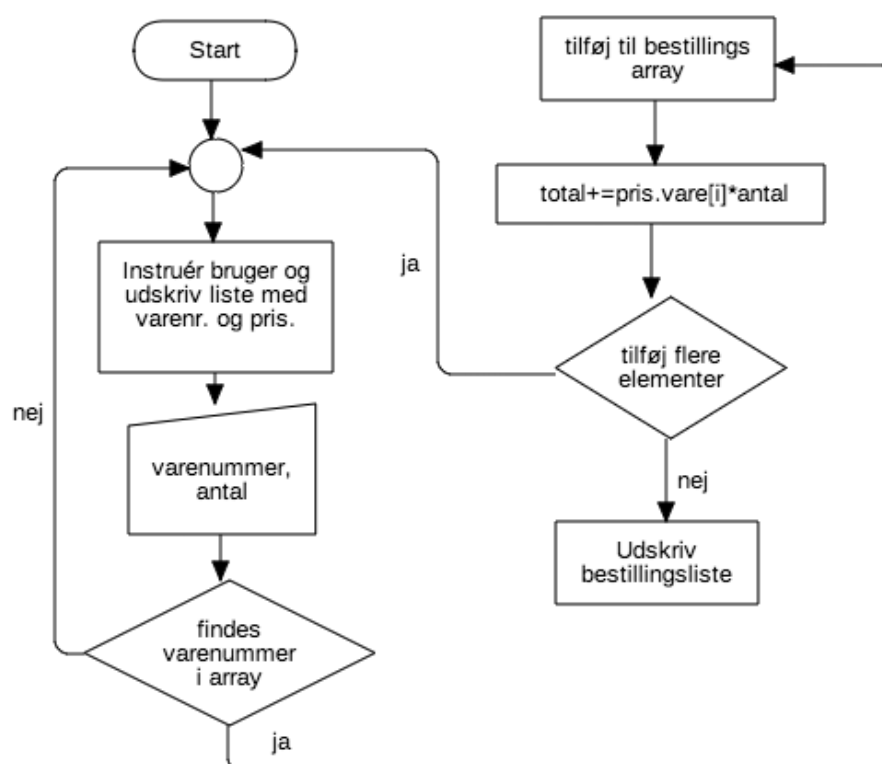
Grundlæggende funktionalitet (opgave 1)

I alle programmer er det vigtigt at få kernen af problemet til at fungere først. I dette tilfælde omhandler det at udskrive en liste over tilgængelige produkter, og tage imod en bestilling, samt beregne en pris.

Hver gang der er noget med en liste af ... tænker jeg straks på arrays. Når det så er en liste af enheder med forskellige egenskaber, så som varenr. beskrivelse og pris, tænker jeg structs. Dette giver mulighed for at konstruere en datatype der samler alt information om en enhed i samme variabel. Udover den strukturelle fordel, gør structs også koden mere selvforklarende. Nedenfor er vist den endelige initialisering af struct lagervare.

```
struct lagervare firmaLager[] = {{.vareNr = 1, .pris = 27, .beskrivelse = "Panama papers"},  
    {.vareNr = 2, .pris = 50, .beskrivelse = "Hillary Clintons email server"},  
    {.vareNr = 3, .pris = 203, .beskrivelse = "Donald trumps tax returns"},  
    {.vareNr = 4, .pris = 12, .beskrivelse = "Proof tha the moonlanding was a hoax"},  
    {.vareNr = 5, .pris = 1, .beskrivelse = "North Korean nuklear launch codes"},  
    {.vareNr = 6, .pris = 42, .beskrivelse = "The missing link"},  
    {.vareNr = 7, .pris = 1337, .beskrivelse = "Doughnut holes"},  
    {.vareNr = 8, .pris = 13, .beskrivelse = "sonic screwdriver"},  
    {.vareNr = 9, .pris = 78, .beskrivelse = "The lost ark"},  
    {.vareNr = 10, .pris = 420, .beskrivelse = "Rubber docky"},  
    {.vareNr = 11, .pris = 9999, .beskrivelse = "The lost nazi treasure"}};
```

Nu hvor det er givet på forhånd at structs vil være en god idé, designs et flowchart.



IO streams (opgave 2)

Det er ikke altid meningsfuldt at skulle indtaste eksempeldata hele tiden, derfor er IO stream super smarte. Herunder vises at programmet kaldes i terminalen

```

Total:945kr
[Davids-MBP-2:uge4 dkcookie$ ./uge4<input.txt>output.txt
[Davids-MBP-2:uge4 dkcookie$ cat output.txt
```

Varelager:

varenummer	Pris	Beskrivelse
1	27 kr	Panama papers
2	50 kr	Hillary Clintons email server
3	203 kr	Donald trumps tax returns
4	12 kr	Proof tha the moonlanding was a hoax
5	1 kr	North Korean nuklear launch codes
6	42 kr	The missing link
7	1337 kr	Doughnut holes
8	13 kr	sonic screwdriver
9	78 kr	The lost ark
10	420 kr	Rubber ducky
11	9999 kr	The lost nazi treasure

Varenummer:
Antal:
Vil du tilføje flere varer? (1/0)

input.txt

2
4
1
3
3
1
4
3
1
2
2
0

output.txt

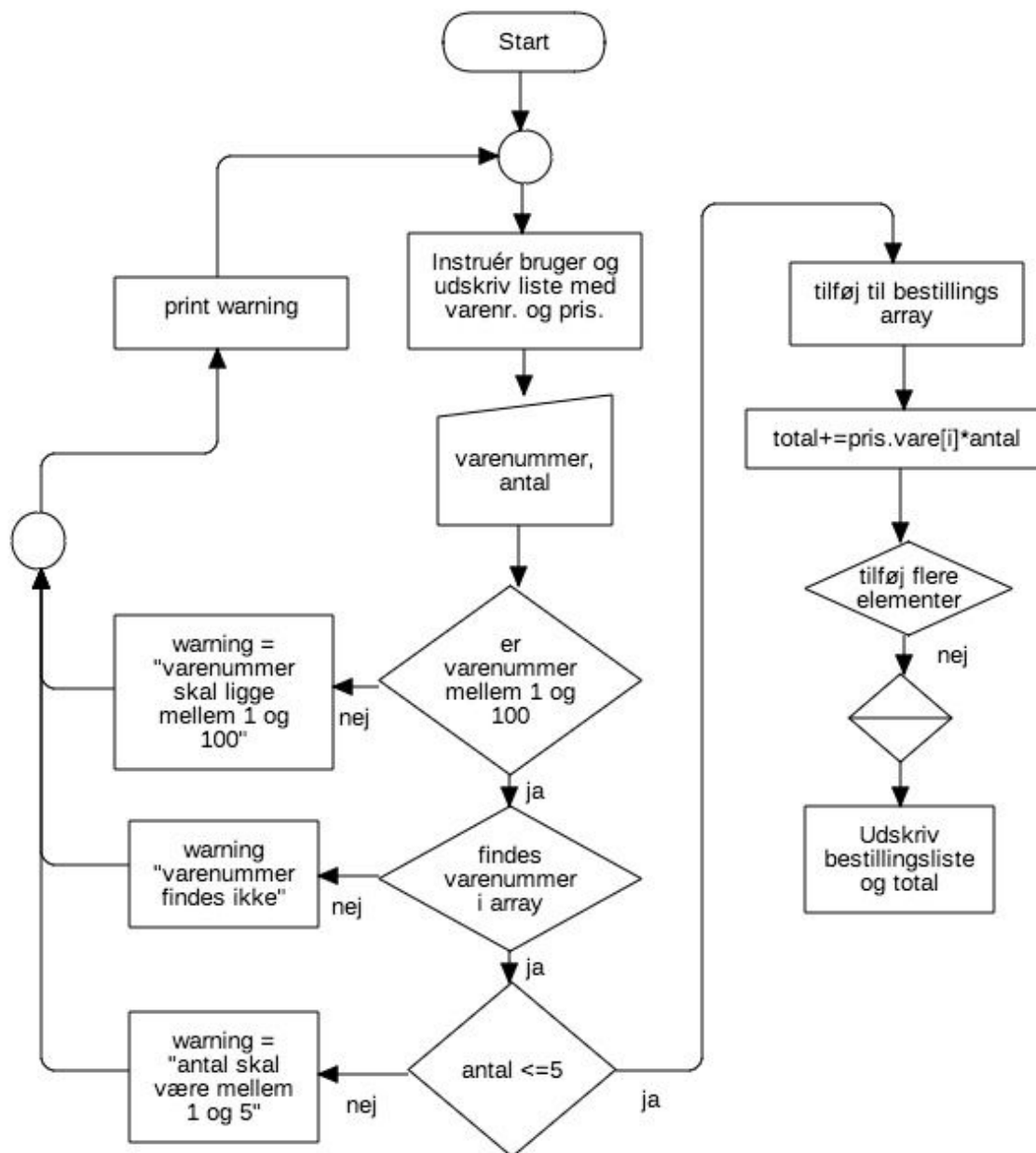
Varelager:

varenummer	Pris	Beskrivelse
1	27 kr	Panama papers
2	50 kr	Hillary Clintons email server
3	203 kr	Donald trumps tax returns
4	12 kr	Proof tha the moonlanding was a hoax
5	1 kr	North Korean nuklear launch codes
6	42 kr	The missing link
7	1337 kr	Doughnut holes
8	13 kr	sonic screwdriver
9	78 kr	The lost ark
10	420 kr	Rubber ducky
11	9999 kr	The lost nazi treasure

Varenummer:
Antal:
Vil du tilføje flere varer? (1/0)
Varenummer:
Antal:
Vil du tilføje flere varer? (1/0)
Varenummer:
Antal:
Vil du tilføje flere varer? (1/0)
Varenummer:
Antal:
Vil du tilføje flere varer? (1/0)

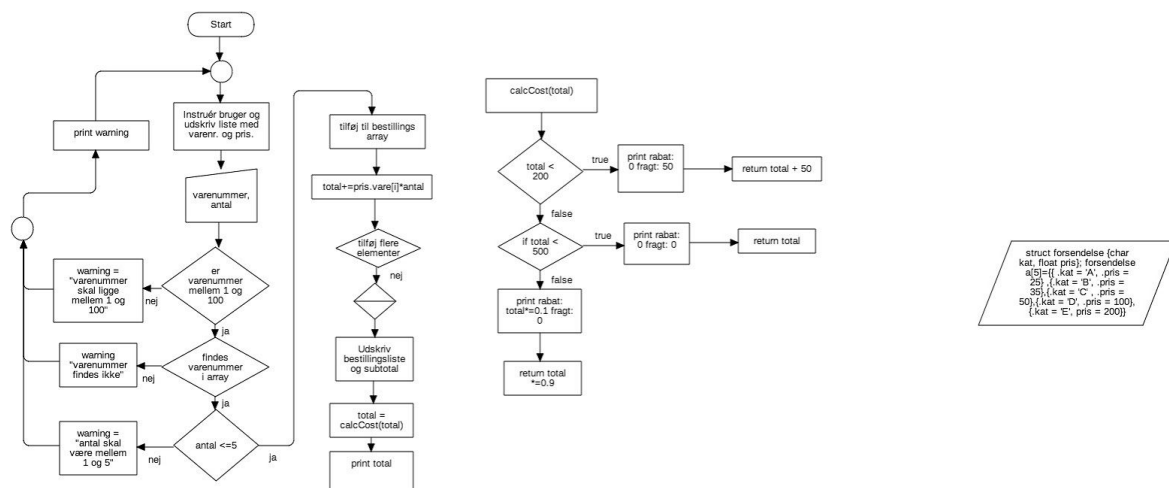
Sortering og Validering (opgave 3)

Sortering sker ved hjælp af en såkaldt bubblesort algoritme. Algoritmen sammenligne parvist elementerne i et array, er det sidste element mindre end det første, bytter elementerne plads. Således løbes arrayet igennem flere gange indtil arrayet er sorteret. Logikken er ellers næsten identisk med opgaven fra uge 2 og der henvises derfor til denne.



Fragt, rabat og en stakkels struct (opgave 4)

I opgave 4 skal total vurderes efter betingelserne angivet i formål og krav. I opgaveformulering beskrives at fragt skal koste 50 kr. Derudover beskrives af der skal oprettes et array med 5 fragtkategorier, men fragt er allerede defineret til 50 kr, så som sagt, så gjort.



De enkelte begrænsninger efterprøves, og valideres at de virker. Ligeledes tjekkes at rabat, forsendelse, samt total og subtotal er korrekt.

Appendix opgave 1

```
/*
Varebestillings software opgave 2.1
Gruppe "taem awesome" (ja det er stavet forkert med vilje)
s153460 Jonas Ladefoged Holm
s113070 David Bjerre Bjørklund
s164920 Markus Visvaldis Ingemann Thieden

Inkluderer relevante libraries
*/
#include "uge41.h"

/*vores main function*/
int main()
{
    struct kundeordre cur_ordre[maksBest];
    int counter = 0;
    int flereVarer = 1;
    int ordreCheck;
    int ordreCheckHolder;

    print_lager(firmaLager);

    while (flereVarer == 1 && counter < maksBest)
    {
        ordreCheck = 1;

        while (ordreCheck == 1) {
            printf("Varenummer:\n");
            scanf("%d", &ordreCheckHolder);

            if (ordreCheckHolder > 0 && ordreCheckHolder <= maksBest) {
                cur_ordre[counter].vareNr = ordreCheckHolder;
                ordreCheck = 0;
            } else {
                printf("Prov igen\n");
            }
        }

        printf("Antal:\n");
        scanf("%d", &cur_ordre[counter].antal);

        counter++;

        printf("Vil du tilføje flere varer? (1/0)\n");

        int manglerSvar = 1;

        while (manglerSvar)
        {
            int svar;
            scanf("%d", &svar);
            switch (svar)
```


Page 2 of 3

}

Appendix opgave 3

```
/*
Varebestillings software opgave 2.1
Gruppe "taem awesome" (ja det er stavet forkert med vilje)
    s153460 Jonas Ladefoged Holm
    s113070 David Bjerre Bjørklund
    s164920 Markus Visvaldis Ingemann Thieden

Inkluderer relevante libraries
*/
#include "uge41.h"

/*vores main function*/
int main()
{
    struct kundeordre cur_ordre[maksBest];
    int counter = 0;
    int flereVarer = 1;
    int ordreCheck;
    int ordreCheckHolder;

    print_lager(firmaLager);

    while (flereVarer == 1 && counter < maksBest)
    {
        ordreCheck = 1;

        while (ordreCheck == 1) {
            printf("Varenummer:\n");
            scanf("%d", &ordreCheckHolder);

            if (ordreCheckHolder > 0 && ordreCheckHolder <= maksBest) {
                cur_ordre[counter].vareNr = ordreCheckHolder;
                ordreCheck = 0;
            } else {
                printf("Prov igen\n");
            }
        }

        ordreCheck = 1;

        while (ordreCheck == 1) {
            printf("Antal:\n");
            ordreCheckHolder;
            scanf("%d", &ordreCheckHolder);
            if (ordreCheckHolder > 0 && ordreCheckHolder < 6) {
                cur_ordre[counter].antal = ordreCheckHolder;
                ordreCheck = 0;
            } else {
                printf("Prov igen\n");
            }
        }

        counter++;
    }
}
```

```
// Gennem søger et array vha linear search and returns index
int lager_search(int varenummer)
{
    for (int i = 0; i < maksBest; i++)
    {
```

}

Appendix opgave 4

```
/*
Varebestillings software opgave 2.1
Gruppe "taem awesome" (ja det er stavet forkert med vilje)
s153460 Jonas Ladefoged Holm
s113070 David Bjerre Bjørklund
s164920 Markus Visvaldis Ingemann Thieden

Inkluderer relevante libraries
*/
#include "uge41.h"

struct forsendelse
{
    char kat;
    float pris;
};
struct forsendelse a[5] = {{.kat = 'A', .pris = 25},
                           {.kat = 'B', .pris = 35},
                           {.kat = 'C', .pris = 50},
                           {.kat = 'D', .pris = 100},
                           {.kat = 'E', .pris = 200}};

/*vores main function*/
int main()
{
    struct kundeordre cur_ordre[maksBest];
    int counter = 0;
    int flereVarer = 1;
    int ordreCheck;
    int ordreCheckHolder;

    print_lager(firmaLager);

    while (flereVarer == 1 && counter < maksBest)
    {
        ordreCheck = 1;

        while (ordreCheck == 1) {
            printf("Varenummer:\n");
            scanf("%d", &ordreCheckHolder);

            if (ordreCheckHolder > 0 && ordreCheckHolder <= maksBest) {
                cur_ordre[counter].vareNr = ordreCheckHolder;
                ordreCheck = 0;
            } else {
                printf("Prov igen\n");
            }
        }

        ordreCheck = 1;

        while (ordreCheck == 1) {
            printf("Antal:\n");
            ordreCheckHolder;
            scanf("%d", &ordreCheckHolder);
            if (ordreCheckHolder > 0 && ordreCheckHolder < 6) {
```


Page 2 of 4

Page 3 of 4

```
        total += 50;
        fors += 50;
    } //end of else if

    //Det fulde resultat, dvs rabat, forsendelse osv skrives
    printf("\n\t\tforsendelse:%d\n", fors);
    printf("\n\t\ttrabat:%.2f\n", rabat);
    printf("\n\t\ttotal pris:%.2f\n", total);
}
```

Appendix Fælles header-fil

```
/*
Varebestillings software opgave 2.1
Gruppe "taem awesome" (ja det er stavet forkert med vilje)
    s153460 Jonas Ladefoged Holm
    s113070 David Bjerre Bjørklund
    s164920 Markus Visvaldis Ingemann Thieden

Inkluderer relevante libraries
*/
#include<stdio.h>
#include<stdlib.h>
#define maksBest lager_len
struct lagervare
{
    int vareNr;
    int pris;
    char* beskrivelse;
};

struct kundeordre
{
    int vareNr;
    int antal;
};

struct lagervare firmaLager[] = {{.vareNr = 1, .pris = 27, .beskrivelse =
    "Panama papers"},
    {.vareNr = 2, .pris = 50, .beskrivelse = "Hillary
        Clintons email server"},
    {.vareNr = 3, .pris = 203, .beskrivelse = "Donald trumps
        tax returns"},
    {.vareNr = 4, .pris = 12, .beskrivelse = "Proof tha the
        moonlanding was a hoax"},
    {.vareNr = 5, .pris = 1, .beskrivelse = "North Korean
        nuklear launch codes"},
    {.vareNr = 6, .pris = 42, .beskrivelse = "The missing
        link"},
    {.vareNr = 7, .pris = 1337, .beskrivelse = "Doughnut
        holes"},
    {.vareNr = 8, .pris = 13, .beskrivelse = "sonic
        screwdriwer"},
    {.vareNr = 9, .pris = 78, .beskrivelse = "The lost ark"},
    ,
    {.vareNr = 10, .pris = 420, .beskrivelse = "Rubber
        docky"},
    {.vareNr = 11, .pris = 9999, .beskrivelse = "The lost
        nazi treasure"}};

//længden af lager_len beregnet ved størrelse i bit / størrelsen af et
//element i bit
int lager_len = sizeof(firmaLager)/sizeof(firmaLager[0]);

void print_lager(struct lagervare lager[]);
```

```
void print_ordre(struct kundeordre ordre[], int count);  
int lager_search(int varenummer);
```