Programmering af beregning til matematik

Matematik D-niveau

Oliver Salvesen – Hold 4504g2dak – ZBC Ringsted

Underviser: Kenn Rydiander Baes

Afleveringsdato: 26.9.2025 eksamensdato: 3.10.2025

Indhold

[Beskrivelse af opgave og det procesfaglige – Problemformulering 2](#__RefHeading___Toc627_2873027200)

[Matematik Teori 2](#__RefHeading___Toc629_2873027200)

[Konstruktion af lommeregnere 2](#__RefHeading___Toc631_2873027200)

[Beregninger 3](#__RefHeading___Toc633_2873027200)

[Promt 3](#__RefHeading___Toc635_2873027200)

[Konklusion 4](#__RefHeading___Toc637_2873027200)

[Refleksion 4](#__RefHeading___Toc639_2873027200)

# Beskrivelse af opgave og det procesfaglige – Problemformulering

1. Hvordan kan jeg programmere nogle simple lommeregnere, som kan hjælpe med procentregning og kvardratrodsregning?
2. Hvad kan jeg bruge dette program til?

Det kan hjælpe mig med at regne procentregning og kvardratrod

1. Hvilke funktioner har opstillingen og hvordan virker den?

# Matematik Teori

(præsenter og redegør for relevant matematisk teori – formler og regneregler)

for at regne procenten fra et tal laver jeg udregningen result = base \* percent \* 0.01;

hvilket vil sige basetallet gange procenten gange 0,01(eller divideret med 100(computere er en smule hurtigere til at gange end at dividere)) giver resultatet

til moms er udregningen i koden

result = value \* 1.25;

100 kr \* 1,25 = 125

og

result = value \* 0.80;

100 \* 0,80 = 80

den første tilføjer moms til prisen og den anden fjerner momsen. Grunden til det er 1,25 og 0,80 er fordi at hvis du tager de 0,25 har du en kvart, når du så har 1,25 har du altså 5 kvarte og det oprendlige tal var 4 kvarte så hvis du laver det til en brøk er det 4/5 hvilket er det samme som 0,80

for kvartdratrod er udregningen

result = pow(x, 0.5);

eller matematisk skrevet x^0,5 = resultatet

9^0,5=3

# Konstruktion af lommeregnere

Min lommeregner kan tage procenter af tal, moms og og kvartdratsrods regnere  
så hvis du står i en butik og skal regne moms ud kan du det, eller hvis ved du har en kvardrat kan du både finde sidelængden fra arial og omvændt

de er relativt simple så hvis jeg var kommet hurtigere i gang hville jeg kunne ha lavet mange flere uden problemer

til moms er den relavante kode

case 1:

result = value \* 1.25;

printf("\n%g with vat is %g\n", value, result);

break;

case 2:

result = value \* 0.80;

printf("\n%g without vat is %g\n", value, result);

break;

case er et tegn så hvis du trykker 1 køre den første del og 2 køre den anden del, lad os bruge den første som eksempel til at forklare hvis du skriver for eksempel skriver 1000 spørger programmet om du vil fjerne eller tilføje moms, du trykker så 1 og programmet skriver så så med printf "1000 with vat is 1250" vat er det engelske ord for moms, jeg foretrækker at kode på engelsk da det kan hjælpe andre i det at jeg har det på github så hvis nu du skulle finde ud af hvordan folk gør noget af det jeg har gjort i min kode og du ikke kunne dansk kan du stadig forstå min kode og dermed bruge dele af det i din egen kode

# Beregninger

Jeg har lavet i alt 5 lommeregnere som jeg lige gennemgår

den første regner en procent af et tal ved hjælp af x \* % / 100

den anden regner et helt tal fra en procent ved hjælp af x \* 100 / %

den tredje tager enten x \* 1,25 for at tilføje eller x \* 0,80 for at fjerne moms

den fjerde tager tager x^0,5 for at regne kvartdraten ud

og den femte og sidste regner kvartdraten af et tal via x \* x hvilket jo er det samme som x^2

# Prompt

Promt til første lommeregner: good now make the first program it has to ask me for a number then it has to ask me for a number then a procentage and output that procentage of that number

promt til anden lommeregner: okay so i need a program where i input a number it will then ask me to choose 1 or 2 where 1 multiplies by 1.25 and 2 multiply by 0.80

promt til tredje lommeregner: write a simple program where you give it a number and it finds the square root of the number

og den sidste lavede jeg i hånden og brugte kun ai til debugging

# Konklusion

Projektet har vist, at det er både muligt og praktisk at udvikle enkle lommeregnere, som håndterer procentberegning, momsjustering, kvadratrods‑ og kvadrats‑beregning. De fire programmer demonstrerer, hvordan grundlæggende matematiske formler kan implementeres med få linjer kode, og hvordan brugerinteraktion (valg via case‑statement) gør dem anvendelige i hverdags‑situationer som indkøb eller byggearbejde.

Den største styrke ligger i den modulære opbygning: hver funktion (procent, moms, kvadratrod, kvadrat) kan udvides eller kombineres uden større omskrivning. Samtidig viser brugen af engelske kommentarer og struktureret kode, at projektet er let at dele på platforme som GitHub, hvilket fremmer genanvendelighed og samarbejde.

Fremadrettet kan flere funktioner (fx eksponentiel beregning eller valutakonvertering) tilføjes, og brugerfladen kan forbedres med grafisk interface eller fejlhåndtering, men de nuværende resultater bekræfter, at simple matematiske værktøjer kan bygges hurtigt og effektivt med grundlæggende programmeringskoncepter.

# Refleksion

At arbejde med dette projekt har givet mig indsigt i, hvordan grundlæggende matematiske operationer kan omsættes til funktionel kode med minimal kompleksitet. Flere centrale erfaringer fremkom:

De fire lommeregnere er korte og har kun en en funktion. Ved at holde hver funktion isoleret (procent, moms, kvadratrod, kvadrat) undgik jeg unødvendig kompleksitet og gjorde koden lettere at vedligeholde. Samt nemmere at finde det stykke der skal endres hvis noget er galt i koden

Implementeringen af et switch/case‑system giver brugeren en intuitiv måde at vælge en udregner. Det viser, hvor vigtigt det er at tænke på brugerens flow, især i simple CLI‑programmer.

Dokumentation og deling; At skrive kommentarer på engelsk og strukturere koden i et offentligt repository på GitHub øger genanvendeligheden. Det gør det også lettere for andre (og mig selv i fremtiden) at forstå og videreudvikle projektet.

Muligheder for udvidelse - Grundstrukturen kan udvides med flere matematiske funktioner (fx eksponentialfunktion, logaritmer) eller med en grafisk brugerflade, hvilket ville gøre værktøjet mere anvendeligt i forskellige kontekster.

Sammenfattende har projektet bekræftet, at selv simple programmer kan levere reel værdi, når de er veldefinerede og let tilgængelige. Refleksionen peger på, at fremtidig hvis jeg ville udvikle det videre skulle jeg nok fokusere på brugeroplevelse, fejlhåndtering og udvidelse af funktionaliteten, så lommeregnerne kan anvendes i et bredere spektrum af praktiske situationer.