

## Du har afleveret en eksamensopgaven i Imperativ Programmering

Hvis du gen-afleverer opgaven overskrives den besvarelse, du netop har fremsendt.

Du kan altid se alle dine afleverede programmer og evt. feedback til disse i [mine afleverede programmer](#)

*Det anbefales at du gemmer eller udskriver denne side som dokumentation for, at du har afleveret eksamensopgaven.*

Her er det afleverede materiale:

Navn: Thomas Bjeldbak Madsen

Afleveringsopgave nummer 5

Kode: 1635249020

Afleveret: October 26, 2021, 13:50:20

E-mail: [tbma21@student.aau.dk](mailto:tbma21@student.aau.dk)

Gruppe: -

Retning: software

Tidsforbrug på opgaven: 360 minutter

Egen vurdering af udbytte: 8

Angivet status: Jeg kunne løse opgaven

Jeg har tilladt at programmet anvendes anonymt i undervisningen og/eller på kursets web sider

Det afleverede program:

```
/*
 * Eksamensopgave 1: En regnemaskine på kommandolinjen
 *
 * Programmer: Thomas Bjeldbak Madsen    Date completed: 26/10/2021
 * Email: tbma21@student.aau.dk         Group: B219 - Software
 * Instructor: Kurt Nørmark              Class: IMPR (DAT1, SW1, IxD3) - AAL E21
 *
 * This program creates a simple calculator.
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void do_next_op(double *accumulator, double operand, char operator);
void scan_data(double *operand, char *operator);
double run_calculator();
int binary_function(char operator);

int main()
{
    run_calculator();

    return (0);
}

/* run_calculator
Her bliver scan_data og do_next_op funktionerne samlet, for at køre udregningen og outputtet af programmet. */

double run_calculator()
{
    double accumulator = 0.0, operand;
    char operator;

    do
    {
        scan_data(&operand, &operator);
        do_next_op(&accumulator, operand, operator);

        if (operator != 'q')
            printf("Your result so far is %lf\n", accumulator);

    } while (operator != 'q');

    printf("Your final result is %lf\n", accumulator);

    return (accumulator);
}

/* scan_data
Her indlæser jeg en operator, og en operand hvis denne er binær. Denne funktion har 2 pointere til operand og operator som den outputter. Hvis der anvendes en unær operator returneres operanden blot som 0.0. I denne bliver binary_function brugt til at checked om den er binær eller unær. */

void scan_data(double *operand, char *operator)
{
    printf("Enter operator, and an optional operand: ");
    scanf(" %c", &*operator);

    if (*operator != 'q')
    {
        if (binary_function(*operator))
```

```

    {
        scanf(" %lf", &*operand);
    }
}

```

/\* do\_next\_op

Denne funktion har 3 input parametre, akkumulator, operator og operand. Med disse to input parametre udfører funktionen den påkrævede for den valgte operator i et switch statement. Da akkumulatoren skal bruges både til input og output er denne en pointer. \*/

```
void do_next_op(double *accumulator, double operand, char operator)
```

```

{
    switch (operator)
    {
        case '+':
        {
            *accumulator = *accumulator + operand;
            break;
        }
        case '-':
        {
            *accumulator = *accumulator - operand;
            break;
        }
        case '*':
        {
            *accumulator = *accumulator * operand;
            break;
        }
        case '/':
        {
            *accumulator = *accumulator / operand;
            break;
        }
        case '^':
        {
            *accumulator = pow(*accumulator, operand);
            break;
        }
        case '%':
        {
            *accumulator = -*accumulator;
            break;
        }
        case '!':
        {
            *accumulator = 1 / *accumulator;
            break;
        }
        case '#':
        {
            if (*accumulator >= 0)
            {
                *accumulator = sqrt(*accumulator);
            }

            break;
        }
    }
}

```

/\* binary\_function

Denne funktion er en boolsk funktion, som indlæser operator variablen, checker om den er binær og hvis den er det returnerer den 1 = true, og hvis den er unær bliver den returneret med 0 = false. \*/

```
int binary_function(char operator)
```

```

{
    if (operator == '+' || operator == '-' || operator == '*' || operator == '/' || operator == '^')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

```