

```

/**
 * Denne filen inneholder et system for å opprette brukere, logge inn, se alle brukere og avslutte
 * programmet.
 *
 * Hver bruker blir nummerert, inneholder brukernavn og et kryptert passord. Alt lagres i Arrayer.
 *
 * @file      Prog1001Oblig2.c
 * @date      27. September 2022
 * @version   1
 * @author    Eirik Hanasand
 */

#include <stdio.h>           //Brukes til bl.a. print
#include <ctype.h>           //Brukes til bl.a. toupper
#include <stdbool.h>        //Brukes til bl.a. utsagn
#include <string.h>         //Brukes til bl.a. strcmp

const int STRLEN = 80;      //Maks lengde input
const int MAXBRUKERE = 20; //Maks antall brukere
const int ASCIINRFORST = 33; //Første lovlige ASCII verdi
const int ASCIINRSIST = 126; //Siste lovlige ASCII verdi.
const int MOD = (ASCIINRSIST-ASCIINRFORST); //Modder ift. lovlig intervall
const int CIPHER = 42;     //Roterer tegn 42 posisjoner

/**
 * Hovedprogram:
 * Inneholder en switch statement hvor brukeren kan velge mellom 4 funksjoner (N, L, S, Q):
 * Funksjoner (forklart i detalj nedenfor)
 *     N - Ny bruker
 *     L - Logge inn
 *     S - Se alle brukere
 *     Q - Avslutte programmet
 *
 * Detaljerte forklaringer:
 *
 * N (Ny bruker):
 * Brukes for å opprette ny bruker, her skrives det inn brukernavn, og passord. Passordet krypteres og
 * deretter lagres dette i arrayene brukerNavn og brukerPass.
 *
 * L (Logge inn):
 * Brukes for å logge inn. Sjekker først om det eksisterer brukere, og dersom det gjør det kan brukeren
 * skrive inn brukernavn og passord.
 *
 * S (Se alle brukere):
 * Viser alle brukere som eksisterer i systemet, feilmelding om det ikke finnes brukere.
 *
 * Q (Avslutte programmet):
 * Avslutter programmet ved å bryte switchen, og deretter gå til slutten av while løkken, som så
 * avslutter programmet ved hjelp av return 0;
 */

int main(){

    //Arrayer
    char brukerNavn[MAXBRUKERE][STRLEN]; //Array for brukere

```

```

char brukerPass[MAXBRUKERE][STRLEN]; //Array for passord
char navnLogin[STRLEN];                //Brukernavn ved innlogging
char passLogin[STRLEN];                //Passord ved innlogging

//Oversikt
char valg;                             //Hva brukeren ønsker å gjøre
int antBrukere = 0;                    //Antall brukere i systemet
int brukerNr = 0;                     //Brukernr innlogget
bool harMellomrom = false;             //Sjekker for mellomrom

//Brukes til passordet
int harStor = 0;                       //Antall store bokstaver
int harLiten = 0;                     //Antall små bokstaver
int harTegn = 0;                       //Antall tegn
int ulovligTegn = 0;                   //Antall ulovlige tegn
bool godkjent = false;                 //Passordets gyldighet
int verdi;                             //ASCII verdi ved kryptering

//Brukes ved innlogging
bool brukerMatch = false;              //Sjekker om brukernavn finnes
bool passMatch = false;                 //Sjekker om passord finnes

do{
    printf("Hva vil du gjøre? (N, L, S, Q)\n"); //Spør brukeren om valg
    gets(&valg);                               //Henter brukerens valg
    valg = toupper(valg);                       //Store bokstaver

    //Inneholder alle 4 valgene brukeren har
    switch (valg) {

        case 'N': //Opprett ny bruker

            //Sjekker om det er plass til flere brukere
            if (antBrukere < MAXBRUKERE) {

                do{ //Looper til brukernavn er lovlig

                    harMellomrom = 0;           //Antall mellomrom
                    printf("Velg brukernavn:\n"); //Ber om brukernavn
                    gets(brukerNavn[antBrukere]); //Henter brukernavn

                    //Sjekker for tomt brukernavn
                    if (strlen(brukerNavn[antBrukere]) == 0) {
                        printf("Brukernavn kan ikke være tomt.\n");

                        //Øker harMellomrom (sjekkes senere)
                        harMellomrom++;
                    }

                    //Sjekker for mellomrom i brukernavnet
                    if (strchr(brukerNavn[antBrukere], ' ') != 0){
                        printf("Mellomrom ikke tillatt!\n");

                        //Øker harMellomrom (sjekkes senere)
                        harMellomrom++;
                    }
                }
            }
        }
    }
}

```

```

    }

    //Looper til brukernavn er lovlig
    }while(harMellomrom > 0);

do{      //Til passord tilfredsstiller alle krav

    //I tilfelle flere runder
    harMellomrom = 0;      //Antall mellomrom
    harStor = 0;           //Antall store bokstaver
    harLiten = 0;          //Antall små bokstaver
    harTegn = 0;           //Antall tegn
    ulovligTegn = 0;       //Antall ulovlige tegn

do{ //Looper til passord finnes og ikke har
    mellomrom

        harMellomrom = 0;           //Antall mellomrom
        printf("Velg passord:\n");  //Ber om
        brukernavn
        gets(brukerPass[antBrukere]); //Henter
        brukernavn

        //Sjekker for tomt passord
        if (strlen(brukerPass[antBrukere]) == 0) {
            printf("Passord kan ikke være tomt.\n");

            //Øker harMellomrom (sjekkes senere)
            harMellomrom++;
        }
        //Sjekker for mellomrom i passordet
        if (strchr(brukerPass[antBrukere], ' ') != 0){
            printf("Mellomrom ikke tillatt!\n");

            //Øker harMellomrom (sjekkes senere)
            harMellomrom++;
        }

        //Looper til passord finnes og ikke har
        mellomrom
    }while(harMellomrom > 0);

//Looper gjennom alle tegn i passordet
for (int i = 0; i <= strlen(brukerPass[antBrukere]
                           )-1; i++) {

    //Sjekker at alle tegn er lovlig ift ASCII
    if(ASCIINRFORST <= (brukerPass[antBrukere][i])
        &&
        ASCIINRSIST >= (brukerPass[antBrukere][i])){

        //Sjekker for stor bokstav
        if(isupper(brukerPass[antBrukere][i])){
            //Øker harStor (sjekkes senere)
            harStor++;
        }
    }
}

```

```

        //Sjekker for liten bokstav
    }else
        if(islower(brukerPass[antBrukere][i])){
            //Øker harLiten (sjekkes senere)
            harLiten++;

            //Hvis ikke bokstav
        }else{
            //Øker harTegn (sjekkes senere)
            harTegn++;
        }

        //Hvis tegn utfor ASCII intervall
    }else{
        //Øker ulovligTegn (sjekkes senere)
        ulovligTegn++;
    }
}

//Feilmelding hvis ulovlige tegn
if(ulovligTegn-1 == 0){
    printf("Passordet inneholder ulovlige tegn.\n");
    godkjent = false;

    //Feilmelding hvis stor bokstav mangler
}else if (harStor == 0){
    printf("Passordet mangler stor bokstav.\n");
    godkjent = false;

    //Feilmelding hvis liten bokstav mangler
}else if (harLiten == 0){
    printf("Passordet mangler liten bokstav.\n");
    godkjent = false;

    //Feilmelding hvis tegn mangler
}else if (harTegn == 0){
    printf("Passordet mangler et tegn.\n");
    godkjent = false;

    //Bool settes til sant hvis passordet er godkjent
}else{
    godkjent = true;
}

//Loopen går til passordet tilfredsstiller alle krav
}while(godkjent == false);

//Looper gjennom og krypterer hvert tegn i passordet
for (int i = 0; i < strlen(brukerPass[antBrukere]);
    i++) {

    //Trekker fra første ascii og modder med cipher
    verdi = ((int)brukerPass[antBrukere][i]
        - ASCIINRFORST + CIPHER) % MOD;

```

```

        //Plusser på første ascii igjen
        int kryptertTegn = verdi + ASCIINRFORST;

        //Skriver kryptert passord til array brukerPass
        brukerPass[antBrukere][i] = (char)kryptertTegn;
        godkjent = true;
    }

    godkjent = false;
    antBrukere++; //Øker antall brukere

    //Nr og navn på bruker som ble lagd
    printf("Du lagde bruker: %i, Brukernavn: %s\n",
        antBrukere, brukerNavn[antBrukere-1]);

}

else{
    //Feilmelding hvis det ikke er plass til flere brukere
    printf("Det er ikke plass til flere brukere\n");
}

break; //Ferdig å opprette bruker

case 'L': //Logg inn

    //Sjekker om det finnes brukere
    if(antBrukere == 0) {

        //Feilmelding hvis det ikke finnes brukere
        printf("Lag bruker først!\n");

    }

    else{
        //Hvis det finnes brukere
        printf("Brukernavn:\n"); //Ber om brukernavn
        gets(navnLogin); //Henter brukernavn

        printf("Passord:\n"); //Ber om passord
        gets(passLogin); //Henter passord

        //Looper gjennom hvert tegn
        for (int i = 0; i < strlen(passLogin); i++) {

            //Trekker fra første ascii og modder med cipher
            verdi = ((int)passLogin[i] - ASCIINRFORST + CIPHER)
                % MOD;

            //Plusser på første ascii igjen
            int kryptertTegn = verdi + ASCIINRFORST;

            //Skriver kryptert passord til array passLogin
            passLogin[i] = (char)kryptertTegn;
        }

        //Looper gjennom brukere
        for (int i = 0; i < antBrukere; i++) {

```

```

        //Leter etter matchende bruker
        brukerMatch = strcmp(navnLogin, brukerNavn[i]);

        //Leter etter matchende passord
        passMatch = strcmp(passLogin, brukerPass[i]);

        //Hvis match finnes
        if (brukerMatch == 0 && passMatch == 0) {
            brukerNr = i+1;
        }
    }

    //Logger inn hvis innloggingsinformasjon stemmer
    if (brukerNr > 0) {

        //Hvilken bruker som ble innlogget
        printf("Innlogget som bruker: %i, brukernavn: %s\n",
            brukerNr, brukerNavn[brukerNr-1]);

        //Feilmelding om innloggingsinformasjon ikke stemmer
    }else{
        printf("Dette stemmer ikke.\n");
    }
}

break; //Ferdig med innlogging

case 'S': //Se alle brukere

    //Sjekker om det finnes brukere
    if(antBrukere == 0) {

        //Feilmelding hvis det ikke finnes brukere
        printf("Det finnes ikke brukere!\n");

    }else{ //Hvis det finnes brukere
        printf("Viser brukere:\n");

        //Looper gjennom arrayene og viser alle verdier
        for (int i = 0; i < antBrukere; i++) {

            //Skriver ut Brukernr, Brukernavn og Passord
            printf("Bruker: %i, Brukernavn: %s, Passord: %s\n",
                i+1, brukerNavn[i], brukerPass[i]);
        }
    }

    break; //Ferdig med å vise brukere

case 'Q': //Avslutt programmet
    break; //Går til while statement

//Feilmelding om innlest verdi ikke er et valg
default:
    printf("Du skrev \"%c\". Prøv igjen.\n", valg);

```

```
        break;
    }

    while(valg != 'Q');    //Avslutter programmet dersom Q blir tastet inn

    //Informerer brukeren om at programmet ble avsluttet
    printf("Du avsluttet programmet med %c.\n", valg);

    return 0;    //Dersom alt gikk bra
}
```