

```

/***
 * Modifisert skjelett/grunnlag for oblig nr 3 i GrProg, høsten 2025.
 *
 * @file    Oblig3-R.c
 * @author  Kristupas Kaupas
 * @date    26.10.2025
 */

#include <stdio.h>          // printf, scanf
#include <stdbool.h>         // bool
#include <ctype.h>           // toupper
#include <math.h>             // kvadratrot

#define ANTRUTER 9           // Antall ruter på brettet
const int STRLEN = 80;       // Tekstlengde
void nullstillBrett();
bool sjekkBrett(const int n);
bool sjekkVinner();
void skrivBrett();
int spillSpillet();

char gBrett[ANTRUTER];      // Spillebrettet.

/***
 * Hovedprogrammet.
 */
int main() {
    char spiller1[STRLEN];    // Begge spillernes navn.
    char spiller2[STRLEN];
    char nyttSpill;           // Kjøre programmet/spillet EN gang til.
    int vinner;                // Evt. spillernummer som har vunnet.

    do {
        nullstillBrett();

        skrivBrett();

        printf("\n\nNavn på spiller 1: ");    scanf(" %s",&spiller1);getchar();
        printf("Navn på spiller 2: ");    scanf(" %s",&spiller2);getchar();
        printf("\n");

        vinner = spillSpillet();

        if (vinner == 1) printf("\nGratulerer %s!!\n\n", spiller1);
        else if (vinner == 2) printf("\nGratulerer %s!!\n\n", spiller2);
            // 'vinner' == 0:
        else    printf("\nIngen vinner denne gangen dessverre.\n\n");

        printf("\n\nEn ny runde (N/j): ");
        scanf(" %c", &nyttSpill);  nyttSpill = toupper(nyttSpill);
        getchar();

    } while (nyttSpill == 'J');

    return 0;
}

```

```

/***
 * Nullstiller/blanker ut alle brettets ruter.
 */
void nullstillBrett() {
    for(int i =0; i<ANTRUTER; i++){
        gBrett[i] = ' ';
                                //Tømmer alle runter
    }
}

/***
 * Finner ut om et trekk er gyldig eller ei.
 *
 * @param n - Ruten (0-8) det forsøkes å sette en brikke i
 * @return Om det var mulig å sette brikken der (true) eller ei (false)
 */
bool sjekkBrett(const int n) {
    if(n >ANTRUTER){return false;}           //Ugyldig rutenummer
    if(gBrett[n] == ' '){return true;}         //True hvis ruten er ledig
    else{return false;}                        //Input er på X eller 0
}

/***
 * Sjekker om noen har tre på rad i en eller annen retning.
 *
 * @return Om noen har tre på rad (true) eller ei (false) i noen retning
 */
bool sjekkVinner() {
    //sjekker horisontal
    int rutenettLengde = sqrt(ANTRUTER);      //Finner lengde/bredde på rutenett
    int oTeller;                            //Teller for hvor mange 0-er algoritmen har funnet
    int xTeller;                            //---||--- men for X
    bool vinnerFunnet = false;//Settes til true hvis en av spillerene har 3 på rad
    int ruteVerdi;

    //Sjekker horisontal
    for(int y = 0; y < rutenettLengde; y++){
        oTeller = 0;
        xTeller = 0;
        for(int x = 0; x < rutenettLengde; x ++){
            ruteVerdi = gBrett[y * rutenettLengde + x]; //verdi på ruta ved bruk av
                                                // x og y kordinater i
                                                // skrivBrett()

            if(ruteVerdi == 'X'){xTeller++;}           //Teller antall X og 0
            else if(ruteVerdi == '0'){oTeller++;}
        }
        //Sjekker om det er 3 X eller 0 for
        //hver eneste linje (1-3), (4-6), etc på skrivBrett rutenett
        if(oTeller == rutenettLengde || xTeller == rutenettLengde){
            vinnerFunnet = true;
        }
    }

    //sjekker vertikal
    for(int x = 0; x<rutenettLengde; x++){
        oTeller = 0;
        xTeller = 0;
    }
}

```

```

for(int y = 0; y < rutenettLengde; y++) {
    //Sjekker det samme som i horisontal, men nå for
    // kolonner (1-7), (5-8), etc
    ruteVerdi = gBrett[y * rutenettLengde + x];
    if(ruteVerdi == 'X') {xTeller++;}
    else if(ruteVerdi == 'O') {oTeller++;}
}
if(oTeller == rutenettLengde || xTeller == rutenettLengde) {
    vinnerFunnet = true;
}
}

//Sjekker diagonal mot høyre
oTeller = 0;
xTeller = 0;
for(int xy = 0; xy < rutenettLengde; xy++) {
    ruteVerdi = gBrett[xy * rutenettLengde + xy];
    if(ruteVerdi == 'X') {xTeller++;} else if(ruteVerdi == 'O') {oTeller++;}
}
if(oTeller == rutenettLengde || xTeller == rutenettLengde) {
    vinnerFunnet = true;
}
} //Bare en diagonal bevegelse i den
   // retningen, så bare 1 loop brukes

//sjekker diagonal mot venstre
oTeller = 0;
xTeller = 0;
for(int xy = 0; xy < rutenettLengde; xy++) {
    ruteVerdi = gBrett[(rutenettLengde - 1 - xy) * rutenettLengde + xy];
    if(ruteVerdi == 'X') {xTeller++;} else if(ruteVerdi == 'O') {oTeller++;}
}
if (oTeller == rutenettLengde || xTeller == rutenettLengde) {
    vinnerFunnet = true;
}

return vinnerFunnet;
}

/***
 * Skriver ut spillebrettet.
 */
void skrivBrett() {
    int rutenettLengde = sqrt(ANTRUTER);           //finner lengde/bredde på rutnett
    int teller = 0;
    for(int y = 0; y < rutenettLengde; y++) {
        for(int x1 = 0; x1 < rutenettLengde; x1++) { //går gjennom alle tall for y
            printf("---%i---", (y * rutenettLengde + x1) + 1);
            if(x1 == rutenettLengde - 1) {
                printf("-");
                } //legger på - på slutten for pynt
        }
        printf("\n"); //Tallene ferdigprintet. Starter på X og 0
    for(int x2 = 0; x2 < rutenettLengde; x2++) {
        if(x2 == 0) {
            printf(" | %c", gBrett[teller]); //Sentrerer tekst i starten
        }
        else{
            printf(" | %c", gBrett[teller]); //sentrerer tekst i midten
        }
    }
}

```

```

    }
    if(x2 == rutenettLengde -1){
        printf(" | ");                                //Sentrerer tekst på slutten av linja
    }
    teller++;
}
printf("\n");
}

/***
 *  Lar spillerne sette brikker, og finner evt en vinner.
 *
 *  @return Spillernummeret som har vunnet (1, 2), evt. 0 (ingen vinner)
 *  @see sjekkBrett(...)
 *  @see skrivBrett(...)
 *  @see sjekkVinner(...)
 */
int spillSpillet() {
    bool spillerTeller = false;                      //False = spiller 1. True = Spiller 2
    char spillerBokstav[2];
    spillerBokstav[0]='X';
    spillerBokstav[1]='O';
    int ruterIgjen;
    int spillerInput;

    do{
        //sjekk om det er noen plasser igjen
        ruterIgjen = 0;
        for(int i = 0; i<ANTRUTER; i++){
            if(sjekkBrett(i)){ruterIgjen++;}           //Sjekker alle mulige ruter
        }
        if(ruterIgjen == 0){return 0;}                  //Alle ruter tatt. Ingen vinner

        skrivBrett();
        printf("Spiller %i, ditt trekk: ", spillerTeller + 1);
        scanf(" %d", &spillerInput);
        if(!sjekkBrett(spillerInput -1)){              //Ruten allerede opptatt
            printf("Du kan ikke sette din brikke her!\n");
            skrivBrett();
        }
        else{
            gBrett[spillerInput -1] = spillerBokstav[spillerTeller];      //Lagre input
            if(sjekkVinner()){
                skrivBrett();return spillerTeller +1;                   //Stopp hvis spiller vant
            }
            spillerTeller = !spillerTeller;                            //Bytter spiller
        }
    }
    while(!sjekkVinner());
}

```