

```

/**
 * Modifisert skjelett/grunnlag for oblig nr 3 i GrProg, høsten 2025.
 *
 * @file    Oblig3.c
 * @author  Malin Foss, William Eide Seiner & FrodeH, Kristupas Kaupas
 */

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

#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] = ' ';    //Fyller alle mulige plasser med noe annet enn X eller 0
    }
}

/**
 * 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;}                //Input er større enn array
    if(gBrett[n] == ' '){return true;}             //Input er på verken X eller 0
    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 kv = sqrt(ANTRUTER); //Tar kvadratroten for å finne lengde/bredde på rutenett

    //Sjekker horisontal
    for(int y = 0; y < kv; y++){
        int x0 = 0;
        int xX = 0;
        for(int x = 0; x < kv; x++){
            int ruteVerdi = gBrett[y * kv + x];    //Gjør om X og Y til indeks (0-8)
                                                    //i array
            if(ruteVerdi == 'X'){xX ++;}            //Teller antall X og 0
            else if(ruteVerdi == '0'){x0 ++;}
        }
        if(x0 == kv || xX == kv){return true;}    //Sjekker om det er 3 X eller 0 for
                                                    //hver eneste linje (1-3), (4-6) etc
    }

    //sjekker vertikal
    for(int x = 0; x<kv; x++){
        int y0 = 0;
        int yX = 0;
        for(int y = 0; y<kv; y++){
            int ruteVerdi = gBrett[y * kv + x];    //Finner indeks i rutenett
            if(ruteVerdi == 'X'){yX ++;}
            else if(ruteVerdi == '0'){y0 ++;}
                                                    //Sjekker det samme som i
                                                    // horisontal, men nå for
                                                    // kolonner (1-7), (5-8), etc
        }
        if(y0 == kv || yX==kv){return true;}
    }
}

```

```

//Sjekker diagonal mot høyre
int xy0 = 0;
int xyX = 0;
for(int xy = 0; xy < kv; xy++){
    int ruteVerdi = gBrett[xy * kv + xy];          //Beregner indeks for diagonale
                                                    // bevegelser
    if(ruteVerdi == 'X'){xyX++;}else if(ruteVerdi == 'O'){xy0++;}
}
if(xy0 == kv || xyX == kv){return true;}          //Bare en diagonal bevegelse i den
                                                    // retningen, så ingen loop brukes

//sjekker diagonal mot venstre
int yx0 = 0;
int yxX = 0;
for(int xy = 0; xy < kv; xy++){
    int ruteVerdi = gBrett[(kv - 1 - xy) * kv + xy]; //Invers diagonal index
    if(ruteVerdi == 'X'){yxX++;} else if(ruteVerdi == 'O'){yx0++;}
}
if (yx0 == kv || yxX == kv){return true;}

return false;          //Ingen av de 4 metodene fant noe. Ingen vinner enda
}

/**
 * Skriver ut spillebrettet.
 */
void skrivBrett() {
    int kv = sqrt(ANTRUTER);          //finner lengde/bredde på rutnett
    int teller = 0;
    for(int y = 0; y < kv; y++){
        for(int x1 = 0; x1 < kv; x1++){          //går gjennom alle mulige tall for y
            printf("---%i--", (y * kv + x1)+1);
            if(x1 == kv-1){
                printf("-");          //legger på - på slutten for pynt
            }
        }
        printf("\n");          //Tallene ferdigprintet. Starter på X og O
        for(int x2 = 0; x2 < kv; x2++){
            if(x2 == 0){
                printf("|  %c",gBrett[teller]);          //Sentrerer tekst i starten
            }
            else{
                printf("  |  %c",gBrett[teller]);          //sentrerer tekst i midten
            }
            if(x2 == kv -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() {
    while(true){
        bool spiller1Spiller = true;          //Hjelpevariabel hvis sjekkBrett() = false
        bool spiller2Spiller = true;

        //sjekk om det er noen plasser igjen
        int ruterIgjen = 0;
        for(int i = 0; i<ANTRUTER; i++){
            if(sjekkBrett(i)){ruterIgjen ++;}//Sjekker alle mulige ruter
        }
        if(ruterIgjen == 0){return 0;}          //Ingen ruter igjen for spillere å bruke

        //spiller 1
        do{
            skrivBrett();
            int spillerInput;
            printf("Spiller 1, ditt trekk: ");
            scanf(" %d", &spillerInput);
            if(!sjekkBrett(spillerInput -1)){    //Feil hvis input er på opptatt plass
                printf("Du kan ikke sette din brikke her!\n");
                skrivBrett();
            }
            else{
                spiller1Spiller = false;          //Spiller var ferdig med sin input
                gBrett[spillerInput -1] = 'X';    //Lagre input
                if(sjekkVinner()){skrivBrett();return 1;} //Sjekk om spiller vant
            }
        }
        while(spiller1Spiller);

        //sjekk om det er noen plasser igjen
        ruterIgjen = 0;
        for(int i = 0; i<ANTRUTER; i++){
            if(sjekkBrett(i)){ruterIgjen ++;}
        }
        if(ruterIgjen == 0){return 0;}

        //spiller 2
        do{
            skrivBrett();
            int spillerInput;
            printf("Spiller 2, ditt trekk: ");
            scanf(" %d", &spillerInput);
            if(!sjekkBrett(spillerInput -1)){    //Feil hvis input er på opptatt plass
                printf("Du kan ikke sette din brikke her!\n");
                skrivBrett();
            }
            else{
                spiller2Spiller = false;          //Spiller ferdig med sin input
                gBrett[spillerInput -1] = 'O';    //Lagre input
                if(sjekkVinner()){skrivBrett();return 2;} //Sjekk om spiller vant
            }
        }
    }
}

```

```
        while(spiller2Spiller);  
    }  
}
```