

```

/**
 * @file      oblig2-R.c
 * @author    Kristupas Kaupas
 * @date      2025-10-13
 * @brief     Obligatorisk oppgave nummer 2 - Reinnlevering
 * @details   Samler inn data for frisbeegolf baner og presenterer det pent
 */

#include <stdio.h>                                //scanf, print, gets
#include <stdbool.h>                               // booleans
#include <ctype.h>                                 //toupper
#include <string.h>                                //strcpy

const int MAXBANER = 18; ///< Maks antall baner.
const int STRLEN = 100; ///< Maks tekstlengde.
int hentBaneLengde(int antallBaner);
int hentPar();
bool hentBaneOB();

/**
 * @brief Startpunktet for programmet.
 * @details Henter og presenterer informasjon for frisbeegolf baner
 * @return returnerer int 0 etter den var kjørt
 */
int main(){
    char baneBeskrivelse[MAXBANER][STRLEN];        // Beskrivelse av banene.
    int baneLengde[MAXBANER];                      // Banenes lengde (i meter).
    int banePar[MAXBANER];                          // Forventet antall kast for å fullføre banen
                                                    // for hver enkelt bane.

    bool baneOB[MAXBANER];                          // Har banene out of bounds
    int antallBaner;                                // Antall baner pr.nå
                                                    //Setter inn verdiene for bane 1 og 2

    antallBaner = 2;
    baneLengde[0] = 62;
    baneLengde[1] = 94;
    banePar[0] = 3;
    banePar[1] = 3;
    baneOB[0] = true;
    baneOB[1] = false;
    strcpy(baneBeskrivelse[0], "Bane med mange trær og kratt");
    strcpy(baneBeskrivelse[1], "Flatt terreng gjennom hele banen");

    char userInput[100];                            //Input array for menyvalg
    while(true){
        printf("Menyvalg:");
        printf("\n    L - Legge til bane");
        printf("\n    S - Se alle baner");
        printf("\n    Q - Asvlutt");
        printf("\nValg:");
        fgets(userInput, sizeof(userInput), stdin);    //Henter input uten /n
        char userChoise = toupper(userInput[0]);       //Tar første bokstav

        switch(userChoise){
            case 'L':{                                //Legge til bane
                if(antallBaner >= MAXBANER){ //tester om max baner er lagret
                    printf("Du har nådd maks baner");
                    printf(", og kan ikke legge til flere.\n");
                    continue;                        //Hopper tilbake til menyvalg
                }
            }
        }
    }
}

```

```

//Lager midlertidige input variabler for noen av spørsmålene
int tempBaneLengde;
char tempBeskrivelse[STRLEN];

tempBaneLengde = hentBaneLengde(antallBaner);
if(tempBaneLengde == -1){
    continue;        //Bruker ønsker å gå tilbake til menyvalg
}

//Lagrer bane lengde
baneLengde[antallBaner] = tempBaneLengde;

//lagrer data for antall par på banen
banePar[antallBaner] = hentPar();

//lagrer true eller false for om det er OB på banen
baneOB[antallBaner] = hentBaneOB();

//spør om banebeskrivelse
printf("Bane Beskrivelse: ");
fgets(tempBeskrivelse, STRLEN -1, stdin); //input til 99 char
//skriver input til beskrivelse array
strcpy(baneBeskrivelse[antallBaner], tempBeskrivelse);

antallBaner ++;
continue;
}
case 'S':{
    //Se alle baner
    int printPointer; //pointer for while loop
    int parCounter = 0; //teller for totale par
    for(printPointer = 0; printPointer < antallBaner;){
        printf("Bane %i", printPointer +1);
        printf("\n    Lengde: %i", baneLengde[printPointer]);
        printf("\n    Par: %i", banePar[printPointer]);

        //printer "Uten." eller "Med." basert på 0 og 1 i array
        if(baneOB[printPointer]){
            printf("\n    Med OB");
        }
        else{
            printf("\n    Uten OB");
        }
        printf("\n    Beskrivelse: %s\n",
            baneBeskrivelse[printPointer]);
        //Øker "par teller" med nåverende par
        parCounter += banePar[printPointer];
        printPointer ++;
    }
    printf("\nAntall baner: %i", antallBaner);
    printf("\nFor å havne på par brukes %i kast.\n", parCounter);
    continue; //Hopper tilbake til hoved meny
}
case 'Q':{
    //Avslutt
    printf("\n\nProgrammet lukkes!");
    return 0;
}
}
}
return 0;

```

```

}
/**
 * @brief Henter bane lengde for menyvalg L
 * @details Spør bruker om banelengde, og returnerer verdi
 * @param[in] antallBaner
 * @return int bane lengde, eller -1 hvis bruker kansellerer input
 */
int hentBaneLengde(int antallBaner){
    int tempBaneLengde;
    while(true){
        printf("Hvor lang er den %i. banen: ", antallBaner +1);
        scanf("%d", &tempBaneLengde);
        getchar();
        if(tempBaneLengde == 0){                //Bruker ønsker å gå til menyvalg
            printf("\n Avslutter.... \n");
            return -1;                //returnerer -1 slik at vi ender tilbake på menyvalg
        }
        if(tempBaneLengde < 0){                //Stoppe bruker fra å sende inn 0 bane lengde
            printf("Tallet kan ikke være mindre enn null!\n");
            continue;                //Hopp til toppen, og spør om tall på nytt
        }
        return tempBaneLengde;
    }
}

/**
 * @brief Henter bane par for menyvalg L
 * @details Spør bruker om bane par, og returnerer riktig verdi
 * @return int 2-8
 * @note Bare slutter/returns når bruker input er riktig
 */
int hentPar(){
    while(true){
        int tempPar;
        printf("Par på banen (2-8): ");
        scanf("%d", &tempPar);
        getchar();
        if(tempPar < 2 || tempPar > 8){                //sjekker om par er mellom 2 og 8
            printf("%i er ikke godkjent. Godkjente tall er (2-8) \n",tempPar);
            continue; //Hopper tilbake til toppen av loop hvis det er feil input
        }
        return tempPar;
    }
}

/**
 * @brief Henter bane OB for menyvalg L
 * @details Spør bruker om bane ob, og returnerer true eller false
 * @return returnerer bool
 * @note Bare returns når bruker input er riktig
 */
bool hentBaneOB(){
    while(true){
        char tempOB;
        printf("Har banen OB (J/N): ");
        scanf(" %c", &tempOB);
        getchar();
        if(toupper(tempOB) == 'J'){                //Hvis input er J
            return true;
        }
    }
}

```

```
else if(toupper(temp0B) == 'N'){ //Hvis input er N
    return false;
}
else{
    printf("output var %c", temp0B);
    printf("input er verken J eller N \n");
    continue; //Hopper tilbake til toppen av loop
}
}
}
```