

OBLIGER\OBL01.cpp

```
/**
 * Oblig nummer 1
 * Ferdig: Ja
 *
 * @file OBL01.cpp
 * @author Jesper Ruud Soløst
 *
 * @date 30.01.2024
 */

using namespace std;

#include <iostream>           // cout, cin
#include <vector>             // vector
#include <string>             // string
#include "LesData2.h"        // Verktøykasse for lesing av diverse data

struct Restaurant {
    string navn,
    adresse,                 // Gateadresse i en og samme by
    type,                   // Indisk, japansk, thai, italiensk, gresk, ....
    omtale;                 // Lengre evaluering/bekrivelse/vurdering
    int terningkast;        // Verdien 1 - 6
};

void nyRestaurant();        //
void restaurantLesData(Restaurant & restaurant); //
void restaurantSkrivData(const Restaurant* restaurant); // Deklarerer alle
void skrivAlleRestauranter(); // funksjonene
void skrivMeny();          //
void slettAlt();           //

vector <Restaurant*> gRestauranter;    // Alle restaurantene i datastrukturen

/**
 * Hovedprogrammet:
 */
int main(){
    char kommando;

    skrivMeny();
    kommando = lesChar("\nKommando");    ///< Alle restaurantene i datastrukturen

    while (kommando != 'Q') {
        switch (kommando) {
            case 'N': nyRestaurant();      break;
            case 'S': skrivAlleRestauranter(); break;
            default:  skrivMeny();         break;
        }
        kommando = lesChar("\nKommando");
    }

    slettAlt();

    return 0;
}
```

```
}

/**
 * Ny restaurant legges inn i datastrukturen (vectoren 'gRestauranter')
 */
void nyRestaurant(){
    Restaurant* nyRestaurant = new Restaurant;    // Ny restaurant-struct.
    cout << "\nNy restaurant:\n";
    restaurantLesData(*nyRestaurant);            // Alle dens data leses inn.
    gRestauranter.push_back(nyRestaurant);        // Legges bakerst i vectoren.
    cout << "\n\tNy restaurant innlagt har nr." << gRestauranter.size() << '\n';
}

/**
 * Leser inn og fyller en restaurants datamedlemmer
 *
 * @param restaurant - Structen som får sine datamedlemmer innlest/fyllt
 */
void restaurantLesData(Restaurant & restaurant){
    cout << "Restaurant: "; getline(cin, restaurant.navn);
    cout << "Adresse   : "; getline(cin, restaurant.adresse);
    cout << "Type       : "; getline(cin, restaurant.type);
    cout << "Omtale     : "; getline(cin, restaurant.omtale);
    restaurant.terningkast = lesInt("Terningkast:", 1,6);
}

/**
 * Skriver en structs data (peker som parameter)
 *
 * @param restaurant - Peker til structen
 */
void restaurantSkrivData(const Restaurant* restaurant){
    cout << "Restaurant : " << restaurant->navn << "\n"
         << "Type       : " << restaurant->type << "\n"
         << "Omtale     : " << restaurant->omtale << "\n"
         << "Terningkast : " << restaurant->terningkast << "\n";
}

/**
 * Skriver alle restaurantene
 */
void skrivAlleRestauranter(){
    for (int i = 0; i < gRestauranter.size(); i++){
        cout << "\n-----[" << i + 1 << "]"-----\n";
        restaurantSkrivData(gRestauranter[i]);    // Skriver ut restaurant i
    }
}

/**
 * Skriver programmets menyvalg/muligheter på skjermen.
 */
void skrivMeny(){
    cout << "\nFoelgende kommandoer er tilgjengelig:\n"
         << "\tN - Ny restaurant\n"
         << "\tS - Se alle restauranter\n"
         << "\tQ - Quit / avslutt\n";
}

/**
 * Sletter/fjerner ALLE tilpekte structer (Bok), og alle pekerne i vectoren.
 */
```

```
*/  
void slettAlt(){  
    for (int i = 0; i < gRestauranter.size(); i++) // Sletter ALLE bøkene/struct'ene:  
        delete gRestauranter[i];  
    gRestauranter.clear();  
    cout << "\n\n\n\tAlle elementer og pekere er slettet...";  
}
```