**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**ĐỀ TÀI: XÂY DỰNG BÀI TOÁN QUẢN LÝ PHÒNG SỬ DỤNG DANH SÁCH LIÊN KẾT ĐÔI**

**Giáo viên hướng dẫn: TS. Đậu Hải Phong**

**Sinh viên thực hiện:**  1. Nguyễn Thành Nguyên

2. Đỗ Văn Vinh

**Hà Nội, 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**ĐỀ TÀI: XÂY DỰNG BÀI TOÁN QUẢN LÝ PHÒNG SỬ DỤNG DANH SÁCH LIÊN KẾT ĐÔI**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1871020437 | Nguyễn Thành Nguyên | 29/05/2006 |  |  |
| 2 | 1871020660 | Đỗ Văn Vinh | 02/01/2006 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, 2025**

**MỤC LỤC**

**(Đánh tự động với 3 mức)**

1. **Giới thiệu đề tài**

Đề tài Xây dựng bài toán quản lý phòng sử dụng danh sách liên kết đôi là một dự án nghiên cứu và thực tế trong lĩnh vực quản lý khách sạn và lập trình máy tính. Đề tài này tập trung vào việc thiết kế và phát triển một hệ thống quản lý phòng cho khách sạn bằng cách sử dụng cơ chế danh sách liên kết đôi.

Danh sách liên kết đôi (doubly linked list) là một cấu trúc dữ liệu mạnh mẽ trong lập trình, được sử dụng để lưu trữ và quản lý dữ liệu theo dạng danh sách các phần tử liên kết với nhau theo cả hai chiều: tiến và lùi. Trong đề tài này, danh sách liên kết đôi được sử dụng để lưu trữ thông tin về các phòng trong khách sạn, cho phép quản lý linh hoạt và hiệu quả các thao tác như thêm, xóa, sửa, và tìm kiếm phòng.

1. **Phân tích đề tài**
   1. **Dữ liệu lưu trữ**

* Mã phòng – kiểu int
* Tên phòng – kiểu string
* Diện tích – kiểu double
* Số giường – kiểu int
* Giá phòng – kiểu double
* Loại phòng – kiểu string
* Loại giường – kiểu string
* Phòng đã đặt hay chưa – kiểu boolean
  1. **Cấu trúc dữ liệu**

Struct Phong {

int masoPhong; // Ma so phong

string loaiPhong; //Loai phong

double giaPhong; //Gia cua phong

double dientich; //Dien tich cua phong

int soGiuong; // So giuong trong phong

string loaiGiuong; //Giuong don/doi

bool daDat; //Kiem tra xem phong da duoc dat hay chua

Phong\* prev;

Phong\* next;

};

* 1. **Các chức năng**
* Thêm phòng mới
* Kiểm tra phòng
* Hiển thị danh sách phòng
* Đặt phòng
* Chỉnh sửa thông tin phòng
* Xóa phòng
* Lưu danh sách phòng vào file
* Đọc danh sách phòng từ file
* Tìm kiếm phòng theo loại
* Hiển thị phòng có số giường và loại giường cụ thể
* Hủy đặt phòng
* Tìm phòng có giá cao nhất
* Thống kê số lượng phòng
* Hiển thị danh sách phòng đã đặt
* Hiển thị danh sách phòng còn trống
* Tính tổng doanh thu từ các phòng đã đặt
* Sắp xếp phòng theo giá (tăng dần)
* Đặt nhiều phòng cùng lúc
* Xóa toàn bộ danh sách phòng
* Tìm phòng theo mã số phòng
* Hiển thị tổng số lượng phòng
* Sửa giá phòng (nếu có)
* Tìm kiếm phòng theo khoảng giá (từ X đến Y)
* Tìm kiếm phòng theo trạng thái (Đã đặt/Còn trống)
* Hiển thị danh sách phòng dưới dạng bảng
* Tìm phòng có diện tích lớn nhất
* Thống kê số lượng giường theo loại phòng
* Tính tổng số giường trong tất cả các phòng
* Hiển thị số phòng có giường đôi còn trống
* Tính tổng số phòng có giường đôi
* Hiển thị danh sách phòng có giường đôi
* Sắp xếp phòng theo loại phòng (từ A đến Z)
* Hiển thị phòng theo loại phòng
* Sắp xếp phòng theo diện tích (tăng dần)
* Hiển thị phòng theo số giường
* Hiển thị phòng theo giá nhỏ hơn hoặc bằng một giá trị cụ thể
* Thống kê số phòng theo loại giường (Đơn/đôi)
* Tìm phòng theo loại giường (Đơn/đôi)
* Sửa số giường của phòng
* Thoát khỏi chương trình
  1. **Thiết kế menu**

==== MENU QUAN LY PHONG =====

1. Them phong moi

2. Kiem tra phong

3. Hien thi danh sach phong

4. Dat phong

5. Chinh sua thong tin phong

6. Xoa phong

7. Luu danh sach phong vao file

8. Doc danh sach phong tu file

9. Tim kiem phong theo loai

10. Hien thi phong co so giuong va loai giuong cu the

11. Huy dat phong

12. Tim phong co gia cao nhat

13. Thong ke so luong phong

14. Hien thi danh sach phong da dat

15. Hien thi danh sach phong con trong

16. Tinh tong doanh thu tu cac phong da dat

17. Sap xep phong theo gia (tang dan)

18. Dat nhieu phong cung luc

19. Xoa toan bo danh sach phong

20. Tim phong theo ma so phong

21. Hien thi tong so luong phong

22. Sua gia phong (neu co)

23. Tim kiem phong theo khoang gia (Tu X den Y)

24. Tim kiem phong theo trang thai (Da dat/Con trong)

25. Hien thi danh sach phong duoi dang bang

26. Tim phong co dien tich lon nhat

27. Thong ke so luong giuong theo loai phong

28. Tinh tong so giuong trong tat ca cac phong

29. Hien thi so phong co giuong doi con trong

30. Tinh tong so phong co giuong doi

31. Hien thi danh sach phong co giuong doi

32. Sap xep phong theo loai phong (A-Z)

33. Hien thi phong theo loai phong

34. Sap xep phong theo dien tich (tang dan)

35. Hien thi phong theo so giuong

36. Hien thi phong theo gia nho hon hoac bang mot gia tri cu the

37. Thong ke so phong theo loai giuong (don/doi)

38. Tim phong theo loai giuong (Don/doi)

39. Sua so giuong cua phong

40. Thoat

Chon chuc nang:

1. **Phân tích chương trình** 
   1. **Cấu trúc dữ liệu**

* Danh sách liên kết đôi
* Cấu trúc Phong (Node)
  1. **Khai báo thư viện**
* **iostream**: Thư viện này cung cấp các chức năng nhập/xuất chuẩn (input/output), cho phép chương trình giao tiếp với người dùng thông qua bàn phím và màn hình. Ví dụ, cout được sử dụng để in ra màn hình và cin được sử dụng để đọc dữ liệu từ bàn phím.
* **string**: Thư viện này cung cấp lớp string để làm việc với chuỗi ký tự. Nó cho phép bạn tạo, thao tác và quản lý chuỗi một cách dễ dàng.
* **fstream**: Thư viện này cung cấp các lớp để làm việc với tệp tin (file). Nó cho phép bạn đọc và ghi dữ liệu từ/vào tệp tin. Trong mã nguồn của bạn, nó được sử dụng để lưu và đọc danh sách phòng từ tệp tin "danh\_sach\_phong.txt".
  1. **Khởi tạo nút**

1. timPhong(int masoPhong)

* Duyệt qua danh sách liên kết để tìm phòng có mã số masoPhong.
* Nếu tìm thấy, trả về con trỏ đến phòng đó. Nếu không, trả về nullptr.

2. themPhong()

* Nhập thông tin phòng từ người dùng.
* Kiểm tra mã số phòng đã tồn tại chưa, nếu có thì dừng lại.
* Nếu không, tạo đối tượng Phong mới và thêm vào danh sách liên kết.
* Cập nhật con trỏ đầu (dau) và cuối (cuoi) khi thêm phòng.

3. xoaPhong(int soPhong)

* Tìm phòng theo mã số. Nếu không có, in thông báo lỗi.
* Nếu có, cập nhật con trỏ của phần tử trước và sau phòng bị xóa.
* Giải phóng bộ nhớ để tránh rò rỉ dữ liệu.

4. datPhong(int masoPhong)

* Kiểm tra xem phòng có tồn tại không.
* Nếu có, kiểm tra trạng thái đặt phòng:
  + Nếu chưa được đặt, đánh dấu phòng là đã đặt.
  + Nếu đã đặt trước đó, hiển thị thông báo.

5. chinhSuaPhong(int soPhong)

* Tìm phòng theo mã số.
* Nếu tìm thấy, cập nhật thông tin mới cho loại phòng và giá phòng.
* Nếu không tìm thấy, in thông báo lỗi.

6. luuDanhSachPhong()

* Mở file để ghi danh sách phòng.
* Duyệt danh sách liên kết và lưu thông tin từng phòng vào file.
* Đóng file sau khi ghi xong.

7. hienThiPhong()

* Kiểm tra danh sách có rỗng không. Nếu rỗng, in thông báo.
* Nếu có phòng, duyệt danh sách liên kết và in thông tin từng phòng.

8. docDanhSachPhong()

* Mở file để đọc danh sách phòng. Nếu file không tồn tại, in thông báo lỗi.
* Đọc từng dòng và khởi tạo đối tượng Phong.
* Thêm vào danh sách liên kết và cập nhật con trỏ đầu/cuối.
* Gọi hienThiPhong() để hiển thị danh sách sau khi đọc.

9. timPhongTheoLoai(string loai)

* Duyệt danh sách để tìm các phòng có loại phù hợp.
* Nếu tìm thấy, hiển thị thông tin phòng.
* Nếu không tìm thấy, in thông báo lỗi.

10. hienThiPhongTheoSoGiuongVaLoaiGiuong(int soGiuong, string loaiGiuong)

* Kiểm tra danh sách phòng có rỗng không. Nếu rỗng, in thông báo.
* Duyệt danh sách phòng, kiểm tra nếu số giường và loại giường phù hợp với yêu cầu đầu vào.
* In thông tin phòng phù hợp.
* Nếu không có phòng nào khớp, hiển thị thông báo.

11. huyDatPhong(int soPhong)

* Tìm phòng theo mã số.
* Nếu phòng tồn tại và đã được đặt, hủy đặt (daDat = false).
* Nếu phòng chưa được đặt, in thông báo không thể hủy.
* Nếu không tìm thấy phòng, hiển thị thông báo lỗi.

12. timPhongGiaCaoNhat()

* Kiểm tra danh sách phòng có rỗng không. Nếu rỗng, in thông báo.
* Duyệt danh sách để tìm phòng có giá cao nhất.
* Hiển thị thông tin phòng có giá cao nhất.

13. thongKePhong()

* Khởi tạo hai biến đếm số lượng phòng đã đặt (soPhongDat) và phòng còn trống (soPhongTrong).
* Duyệt danh sách phòng, cập nhật giá trị cho hai biến đếm.
* Hiển thị tổng số phòng đã đặt và số phòng còn trống.

14. hienThiPhongDaDat()

* Duyệt danh sách phòng và hiển thị các phòng đã được đặt.
* Nếu không có phòng nào được đặt, hiển thị thông báo.

15. hienThiPhongConTrong()

* Duyệt danh sách phòng và hiển thị các phòng còn trống.
* Nếu không có phòng nào còn trống, hiển thị thông báo.

16. tinhTongDoanhThu()

* Khởi tạo biến tongDoanhThu để tính tổng doanh thu từ các phòng đã đặt.
* Duyệt danh sách, cộng dồn giá phòng nếu phòng đã được đặt.
* Hiển thị tổng doanh thu.

17. sapXepPhongTheoGiaTangDan()

* Kiểm tra nếu danh sách phòng rỗng hoặc chỉ có một phòng thì không cần sắp xếp.
* Sử dụng thuật toán Bubble Sort để sắp xếp danh sách phòng theo giá tăng dần.
* Hoán đổi dữ liệu giữa hai phòng nếu giá phòng trước lớn hơn giá phòng sau.
* Hiển thị thông báo khi sắp xếp xong.

18. datNhieuPhong()

* Nhập số lượng phòng cần đặt.
* Nếu số lượng không hợp lệ, hiển thị thông báo.
* Lặp qua số lượng phòng yêu cầu, nhập mã số phòng từng phòng.
* Nếu phòng tồn tại và chưa được đặt, cập nhật trạng thái daDat = true.
* Nếu phòng không tồn tại hoặc đã được đặt, hiển thị thông báo lỗi.

1. timPhongTheoMaSo(int maSoPhong)

* Chức năng: Tìm phòng theo mã số.
* Cách hoạt động:
  + Gọi timPhong(maSoPhong) để tìm phòng có mã số tương ứng.
  + Nếu tìm thấy, hiển thị thông tin chi tiết của phòng.
  + Nếu không tìm thấy, in thông báo lỗi.

1. demSoLuongPhong()

* Chức năng: Đếm số lượng phòng trong danh sách.
* Cách hoạt động:
  + Duyệt danh sách liên kết bằng con trỏ temp, tăng biến đếm count.
  + Trả về tổng số phòng.

1. xoaToanBoDanhSachPhong()

* Chức năng: Xóa toàn bộ danh sách phòng.
* Cách hoạt động:
  + Kiểm tra nếu danh sách rỗng thì in thông báo.
  + Duyệt danh sách và xóa từng phòng, cập nhật con trỏ dau.
  + Đặt cuoi về nullptr.

1. suaGiaPhong(int maSoPhong, double giaMoi)

* Chức năng: Cập nhật giá phòng dựa trên mã số.
* Cách hoạt động:
  + Kiểm tra nếu giaMoi <= 0, in thông báo lỗi.
  + Gọi timPhong(maSoPhong), nếu phòng tồn tại thì cập nhật giaPhong.
  + Nếu phòng không tồn tại, in thông báo lỗi.

1. timPhongTheoGia(double giaMin, double giaMax)

* Chức năng: Tìm phòng trong khoảng giá nhất định.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra nếu giá phòng nằm trong khoảng [giaMin, giaMax] thì in thông tin.
  + Nếu không tìm thấy phòng nào, in thông báo.

1. timPhongTheoTrangThai(string trangThai)

* Chức năng: Tìm các phòng dựa trên trạng thái (Đã đặt hoặc Còn trống).
* Cách hoạt động:
  + Chuyển đổi trạng thái từ chuỗi thành boolean (true/false).
  + Duyệt danh sách và in thông tin phòng có trạng thái phù hợp.

1. hienThiPhongDangBang()

* Chức năng: Hiển thị danh sách phòng dưới dạng bảng.
* Cách hoạt động:
  + In tiêu đề bảng.
  + Duyệt danh sách, in từng phòng theo định dạng cột.

1. timPhongDienTichLonNhat()

* Chức năng: Tìm phòng có diện tích lớn nhất.
* Cách hoạt động:
  + Duyệt danh sách, giữ phòng có diện tích lớn nhất và cập nhật nếu tìm thấy phòng lớn hơn.
  + In thông tin phòng có diện tích lớn nhất.

1. demSoGiuongTheoLoaiPhong()

* Chức năng: Đếm tổng số giường theo từng loại phòng.
* Cách hoạt động:
  + Duyệt danh sách, cộng tổng số giường cho từng loại phòng.
  + Tránh in trùng loại phòng bằng cách kiểm tra trước khi in.

1. tinhTongSoGiuong()

* Chức năng: Tính tổng số giường của tất cả các phòng.
* Cách hoạt động:
  + Duyệt danh sách và cộng dồn số giường.

1. hienThiPhongGiuongDoiConTrong()

* Chức năng: Hiển thị danh sách các phòng có giường đôi chưa được đặt.
* Cách hoạt động:
  + Duyệt danh sách và kiểm tra điều kiện !temp->daDat và temp->loaiGiuong == Doi.

1. tinhTongSoPhongGiuongDoi()

* Chức năng: Đếm số lượng phòng có giường đôi.
* Cách hoạt động:
  + Duyệt danh sách, đếm số phòng có loaiGiuong == Doi.

1. hienThiPhongGiuongDoi()

* Chức năng: Hiển thị tất cả các phòng có giường đôi.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra loaiGiuong == Doi, nếu đúng thì in thông tin.

1. sapXepPhongTheoLoaiPhong()

* Chức năng: Sắp xếp danh sách phòng theo loại (từ A đến Z).
* Cách hoạt động:
  + Dùng thuật toán Bubble Sort để hoán đổi dữ liệu giữa các node.

1. hienThiPhongTheoLoaiPhong(string loaiPhong)

* Chức năng: Hiển thị tất cả các phòng thuộc loại phòng cụ thể.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra loaiPhong và in thông tin nếu trùng.

1. hienthiPhongTheoSoGiuong(int soGiuongCanTim)

* Chức năng: Hiển thị các phòng có số giường cụ thể.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra soGiuong == soGiuongCanTim.

1. hienthiPhongTheoLoaiGiuong(string loaiGiuongCanTim)

* Chức năng: Hiển thị các phòng có loại giường cụ thể.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra loaiGiuong == loaiGiuongCanTim.

1. sapXepPhongTheoDienTichTangDan()

* Chức năng: Sắp xếp danh sách phòng theo diện tích tăng dần.
* Cách hoạt động:
  + Dùng thuật toán Bubble Sort để hoán đổi dữ liệu.

1. thongKePhongTheoLoaiGiuong()

* Chức năng: Thống kê số lượng phòng theo loại giường (Đơn/Đôi).
* Cách hoạt động:
  + Duyệt danh sách và đếm số phòng có giường đơn và giường đôi.

1. timPhongTheoLoaiGiuong(string loaiGiuong)

* Chức năng: Tìm tất cả các phòng có loại giường cụ thể.
* Cách hoạt động:
  + Duyệt danh sách, kiểm tra loaiGiuong, nếu trùng thì in thông tin.

39. Sửa số giường

Hàm cập nhật số giường của một phòng dựa trên mã số phòng.

* **Tìm phòng** có mã maSoPhong. Nếu không tồn tại, thông báo lỗi và thoát.
* **Kiểm tra số giường mới**:
  + Nếu soGiuongMoi <= 0, thông báo lỗi và thoát.
* **Cập nhật số giường** của phòng và thông báo thành công.

40.Thoát

* Chức năng: Thoát chương trình.

Dưới đây là đoạn code của bài :

*#include <iostream>*

*#include <string>*

*#include <fstream>*

*using namespace std;*

*// Cấu trúc phong*

*struct Phong {*

*int masoPhong; // Ma so phong*

*string loaiPhong; //Loai phong*

*double giaPhong; //Gia cua phong*

*double dientich; //Dien tich cua phong*

*int soGiuong; // So giuong trong phong*

*string loaiGiuong; //Giuong don/doi*

*bool daDat; //Kiem tra xem phong da duoc dat hay chua*

*Phong\* prev;*

*Phong\* next;*

*};*

*// Con tro dau va cuoi danh sach*

*Phong\* dau = nullptr;*

*Phong\* cuoi = nullptr;*

*// Kiem tra phong*

*Phong\* timPhong(int masoPhong) {*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->masoPhong == masoPhong) {*

*return temp;*

*}*

*temp = temp->next;*

*}*

*return nullptr;*

*}*

*//Them phong moi*

*void themPhong() {*

*int masoPhong,soGiuong;*

*string loaiPhong, loaiGiuong;*

*double giaPhong,dientich;*

*Nhap ma so phong: ;*

*cin >> masoPhong;*

*cin.ignore(); // Xoa ky tu xuong dong sau khi nhap masoPhong*

*if (timPhong(masoPhong)) {*

*phong << masoPhong << đã tồn tại! << endl;*

*return;*

*}*

*Nhap loai phong: ;*

*getline(cin, loaiPhong);*

*Nhap gia phong: ;*

*cin >> giaPhong;*

*Nhap dien tich phong( m² ): ;*

*cin >> dientich;*

*Nhap so giuong: ;*

*cin >> soGiuong;*

*Nhap loai giuong (doi/don): ;*

*cin >> loaiGiuong;*

*Phong\* moi = new Phong{masoPhong, loaiPhong, giaPhong, dientich, soGiuong, loaiGiuong, false, nullptr, nullptr};*

*if (!dau) {*

*dau = cuoi = moi;*

*} else {*

*cuoi->next = moi;*

*moi->prev = cuoi;*

*cuoi = moi;*

*}*

*Them phong co ma so: << masoPhong << thanh cong! << endl;*

*}*

*// Xoa phong*

*void xoaPhong(int soPhong) {*

*Phong\* phong = timPhong(soPhong);*

*if (!phong) {*

*Khong tim thay phong co ma so << soPhong << de xoa! << endl;*

*return;*

*}*

*if (phong->prev) {*

*phong->prev->next = phong->next;*

*} else {*

*dau = phong->next;*

*}*

*if (phong->next) {*

*phong->next->prev = phong->prev;*

*} else {*

*cuoi = phong->prev;*

*}*

*delete phong;*

*Xoa phong co ma so << soPhong << thanh cong! << endl;*

*}*

*// Dat phong*

*void datPhong(int masoPhong) {*

*Phong\* phong = timPhong(masoPhong);*

*if (phong) {*

*if (!phong->daDat) {*

*phong->daDat = true;*

*Phong co ma so << masoPhong << da duoc dat thanh cong! << endl;*

*} else {*

*Phong co ma so << masoPhong << da duoc dat truoc do! << endl;*

*}*

*} else {*

*Khong tim thay phong co ma so << masoPhong << ! << endl;*

*}*

*}*

*// Chinh sua thong tin phong*

*void chinhSuaPhong(int soPhong) {*

*Phong\* phong = timPhong(soPhong);*

*if (phong) {*

*cin.ignore(); // Xoa bo dem truoc khi nhap chuoi*

*Nhap loai phong moi: ;*

*getline(cin, phong->loaiPhong);*

*Nhap gia phong moi: ;*

*cin >> phong->giaPhong;*

*Cap nhat thong tin phong co ma so << soPhong << thanh cong! << endl;*

*} else {*

*Khong tim thay phong co ma so << soPhong << ! << endl;*

*}*

*}*

*// Luu danh sach phong vao file*

*void luuDanhSachPhong() {*

*ofstream file(danh\_sach\_phong.txt);*

*if (!file) {*

*Loi khi mo file de ghi! << endl;*

*return;*

*}*

*Phong\* temp = dau;*

*while (temp) {*

*file << temp->masoPhong << << temp->loaiPhong << << temp->giaPhong << << temp->dientich << << temp->soGiuong << << temp->loaiGiuong << << temp->daDat << endl; // Thêm khoảng trắng giữa soGiuong và loaiGiuong*

*temp = temp->next;*

*}*

*file.close();*

*Luu danh sach phong thanh cong! << endl;*

*}*

*// Hien thi danh sach phong*

*void hienThiPhong() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*Phong\* temp = dau;*

*while (temp) {*

*Ma so phong: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << VND , Dien tich: << temp->dientich*

*<< m², Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*temp = temp->next;*

*}*

*}*

*// Đọc danh sach phong tu file*

*void docDanhSachPhong() {*

*ifstream file(danh\_sach\_phong.txt);*

*if (!file) {*

*Khong tim thay file co danh sach phong! << endl;*

*return;*

*}*

*int soPhong,soGiuong;*

*string loaiPhong, loaiGiuong;*

*double giaPhong,dientich;*

*bool daDat;*

*while (file >> soPhong >> loaiPhong >> giaPhong >> dientich >> soGiuong >> loaiGiuong >> daDat) {*

*Phong\* moi = new Phong{soPhong, loaiPhong, giaPhong, dientich, daDat};*

*if (!dau) {*

*dau = cuoi = moi;*

*} else {*

*cuoi->next = moi;*

*moi->prev = cuoi;*

*cuoi = moi;*

*}*

*}*

*file.close();*

*Doc danh sach phong tu file thanh cong! << endl;*

*hienThiPhong();*

*}*

*// Tim kiem phong theo loai*

*void timPhongTheoLoai(string loai) {*

*Phong\* temp = dau;*

*bool found = false;*

*while (temp) {*

*if (temp->loaiPhong == loai) {*

*Ma so phong: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << VND , Trang thai:*

*<< (temp->daDat ? Da dat : Con trong) << endl;*

*found = true;*

*}*

*temp = temp->next;*

*}*

*if (!found) {*

*Khong tim thay phong thuoc loai << loai << ! << endl;*

*}*

*}*

*// Hien thi phong theo so giuong va loai giuong*

*void hienThiPhongTheoSoGiuongVaLoaiGiuong(int soGiuong, string loaiGiuong) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Danh sach phong co << soGiuong << giuong << loaiGiuong << :*

*while (temp) {*

*if (temp->soGiuong == soGiuong && (temp->loaiGiuong == loaiGiuong || (loaiGiuong == đơn && temp->loaiGiuong == Đơn) || (loaiGiuong == đôi && temp->loaiGiuong == Đôi))) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << VND, Dien tich: << temp->dientich << m²,*

*<< Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Không có phòng nào thỏa mãn điều kiện! << endl;*

*}*

*}*

*// Huy dat phong*

*void huyDatPhong(int soPhong) {*

*Phong\* phong = timPhong(soPhong);*

*if (phong) {*

*if (phong->daDat) {*

*phong->daDat = false;*

*Huy dat phong << soPhong << thanh cong! << endl;*

*} else {*

*Phong << soPhong << chua duoc dat nen khong the huy! << endl;*

*}*

*} else {*

*Khong tim thay phong << soPhong << ! << endl;*

*}*

*}*

*// Tim phong co gia cao nhat*

*void timPhongGiaCaoNhat() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*Phong\* temp = dau;*

*Phong\* phongGiaCaoNhat = dau;*

*while (temp) {*

*if (temp->giaPhong > phongGiaCaoNhat->giaPhong) {*

*phongGiaCaoNhat = temp;*

*}*

*temp = temp->next;*

*}*

*Phong co gia cao nhat: << endl;*

*So phong: << phongGiaCaoNhat->masoPhong << endl;*

*Loai: << phongGiaCaoNhat->loaiPhong << endl;*

*Gia: << phongGiaCaoNhat->giaPhong << VND << endl;*

*Dien tich: << phongGiaCaoNhat-> dientich << m² << endl;*

*Trang thai: << (phongGiaCaoNhat->daDat ? Da dat : Con trong) << endl;*

*}*

*// Thong ke so luong phong con trong và da dat*

*void thongKePhong() {*

*int soPhongDat = 0, soPhongTrong = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->daDat) {*

*soPhongDat++;*

*} else {*

*soPhongTrong++;*

*}*

*temp = temp->next;*

*}*

*So phong da dat: << soPhongDat <<*

*So phong con trong: << soPhongTrong <<*

*}*

*// Hien thi danh sach phong da dat*

*void hienThiPhongDaDat() {*

*Phong\* temp = dau;*

*bool found = false;*

*while (temp) {*

*if (temp->daDat) {*

*Ma so phong: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND*

*found = true;*

*}*

*temp = temp->next;*

*}*

*if (!found) {*

*Khong co phong nao da duoc dat! << endl;*

*}*

*}*

*// Hien thi danh sach phong con trong*

*void hienThiPhongConTrong() {*

*Phong\* temp = dau;*

*bool found = false;*

*while (temp) {*

*if (!temp->daDat) {*

*Ma so phong: << temp->masoPhong*

*<< , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << VND*

*<< , Dien tich: << temp->dientich << m²*

*<<*

*found = true;*

*}*

*temp = temp->next;*

*}*

*if (!found) {*

*Khong co phong nao con trong! << endl;*

*}*

*}*

*// Tinh tong doanh thu tu cac phong da dat*

*void tinhTongDoanhThu() {*

*double tongDoanhThu = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->daDat) {*

*tongDoanhThu += temp->giaPhong;*

*}*

*temp = temp->next;*

*}*

*Tong doanh thu tu cac phong da dat: << tongDoanhThu << VND*

*}*

*// Sap xep phong theo gia*

*void sapXepPhongTheoGiaTangDan() {*

*if (!dau || !dau->next) {*

*Danh sach phong trong hoac chi co mot phong, khong can sap xep! << endl;*

*return;*

*}*

*bool swapped;*

*Phong\* temp;*

*Phong\* last = nullptr;*

*do {*

*swapped = false;*

*temp = dau;*

*while (temp->next != last) {*

*if (temp->giaPhong > temp->next->giaPhong) {*

*// Hoan doi du lieu hai phong*

*swap(temp->masoPhong, temp->next->masoPhong);*

*swap(temp->loaiPhong, temp->next->loaiPhong);*

*swap(temp->giaPhong, temp->next->giaPhong);*

*swap(temp->daDat, temp->next->daDat);*

*swapped = true;*

*}*

*temp = temp->next;*

*}*

*last = temp;*

*} while (swapped);*

*Danh sach phong da duoc sap xep theo gia tang dan! << endl;*

*}*

*// Dat nhieu phong cung luc*

*void datNhieuPhong() {*

*int soLuong;*

*Nhap so luong phong can dat: ;*

*cin >> soLuong;*

*if (soLuong <= 0) {*

*So luong phong khong hop le! << endl;*

*return;*

*}*

*for (int i = 0; i < soLuong; i++) {*

*int maSoPhong;*

*Nhap ma so phong thu << (i + 1) << : ;*

*cin >> maSoPhong;*

*Phong\* phong = timPhong(maSoPhong);*

*if (phong && !phong->daDat) {*

*phong->daDat = true;*

*Da dat phong << maSoPhong << thanh cong! << endl;*

*} else {*

*Phong << maSoPhong << khong ton tai hoac da duoc dat! << endl;*

*}*

*}*

*}*

*// Tim phong theo ma so*

*void timPhongTheoMaSo(int maSoPhong) {*

*Phong\* phong = timPhong(maSoPhong);*

*if (phong) {*

*Thong tin phong:*

*Ma so: << phong->masoPhong <<*

*Loai phong: << phong->loaiPhong <<*

*Gia phong: << phong->giaPhong <<*

*Dien tich : << phong->dientich << m²*

*Trang thai: << (phong->daDat ? Da dat : Con trong) <<*

*} else {*

*Khong tim thay phong co ma so << maSoPhong << !*

*}*

*}*

*//Dem xem co bao nhieu phong*

*int demSoLuongPhong() {*

*int count = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*count++;*

*temp = temp->next;*

*}*

*return count;*

*}*

*//Xoa toan bo danh sach phong*

*void xoaToanBoDanhSachPhong() {*

*if (!dau)*

*{*

*cout<<Danh sach phong rong, khong co gi de xoa!<<endl;*

*return;*

*}*

*while (dau) {*

*Phong\* temp = dau;*

*dau = dau->next;*

*delete temp;*

*}*

*cuoi = nullptr;*

*Toan bo danh sach phong da duoc xoa! << endl;*

*}*

*// Sua gia phong*

*void suaGiaPhong(int maSoPhong, double giaMoi) {*

*Phong\* phong = timPhong(maSoPhong);*

*if (giaMoi <= 0) {*

*Gia moi khong hop le! << endl;*

*return;*

*}*

*phong->giaPhong = giaMoi;*

*if (phong) {*

*phong->giaPhong = giaMoi;*

*Cap nhat gia phong thanh cong! << endl;*

*} else {*

*Khong tim thay phong! << endl;*

*}*

*}*

*//Tim phong theo gia*

*void timPhongTheoGia(double giaMin, double giaMax) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->giaPhong >= giaMin && temp->giaPhong <= giaMax) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong nao trong khoang gia nay. << endl;*

*}*

*}*

*// Tim phong theo trang thai*

*void timPhongTheoTrangThai(string trangThai) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*bool trangThaiBool = (trangThai == Da dat);*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->daDat == trangThaiBool) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , Gia: << temp->giaPhong << , Dien tich: << temp->dientich*

*<< m², Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong nao theo trang thai nay. << endl;*

*}*

*}*

*// Hien thi danh sach phong duoi dang bang*

*void hienThiPhongDangBang() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*// Tieu đe bang*

*--------------------------------------------------------- << endl;*

*| Ma so | Loai phong | Gia | Dien tich (m²) | Trang thai | << endl;*

*--------------------------------------------------------- << endl;*

*Phong\* temp = dau;*

*while (temp) {*

*| << temp->masoPhong;*

*| << temp->loaiPhong;*

*| << temp->giaPhong << VND;*

*| << temp->dientich;*

*| << (temp->daDat ? Da dat : Con trong) << | << endl;*

*temp = temp->next;*

*}*

*--------------------------------------------------------- << endl;*

*}*

*// Tim phong co dien tich lon nhat*

*void timPhongDienTichLonNhat() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*Phong\* temp = dau;*

*Phong\* phongDienTichLonNhat = dau;*

*while (temp) {*

*if (temp->dientich > phongDienTichLonNhat->dientich) {*

*phongDienTichLonNhat = temp;*

*}*

*temp = temp->next;*

*}*

*Phong co dien tich lon nhat:*

*Ma so: << phongDienTichLonNhat->masoPhong <<*

*Loai phong: << phongDienTichLonNhat->loaiPhong <<*

*Gia phong: << phongDienTichLonNhat->giaPhong << VND*

*Dien tich: << phongDienTichLonNhat->dientich << m²*

*Trang thai: << (phongDienTichLonNhat->daDat ? Da dat : Con trong) <<*

*}*

*//Thong ke so giuong theo loai phong*

*void demSoGiuongTheoLoaiPhong() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*Phong\* temp = dau;*

*while (temp) {*

*string loaiPhongHienTai = temp->loaiPhong;*

*int tongSoGiuong = 0;*

*Phong\* temp2 = dau;*

*// Đem so giuong cho loai phong hien tai*

*while (temp2) {*

*if (temp2->loaiPhong == loaiPhongHienTai) {*

*tongSoGiuong += temp2->soGiuong;*

*}*

*temp2 = temp2->next;*

*}*

*// Kiem tra xem loai phong duoc in ra chua*

*bool daIn = false;*

*Phong\* temp3 = dau;*

*while (temp3 != temp) {*

*if (temp3->loaiPhong == loaiPhongHienTai) {*

*daIn = true;*

*break;*

*}*

*temp3 = temp3->next;*

*}*

*// In ket qua neu loai phong chua duoc in*

*if (!daIn) {*

*Loai phong: << loaiPhongHienTai << , So giuong: << tongSoGiuong << endl;*

*}*

*temp = temp->next;*

*}*

*}*

*// Tinh tong so giuong trong tat ca cac phong*

*void tinhTongSoGiuong() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*int tongSoGiuong = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*tongSoGiuong += temp->soGiuong;*

*temp = temp->next;*

*}*

*Tong so giuong trong tat ca cac phong: << tongSoGiuong << endl;*

*}*

*void hienThiPhongGiuongDoiConTrong() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Cac phong co giuong doi con trong:*

*while (temp) {*

*if (!temp->daDat && temp->loaiGiuong == Doi|| temp->loaiGiuong == doi) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², So giuong: << temp->soGiuong << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong giuong doi nao con trong! << endl;*

*}*

*}*

*// Tinh tong so phong co giuong doi*

*void tinhTongSoPhongGiuongDoi() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*int tongSoPhongGiuongDoi = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->loaiGiuong == Doi|| temp->loaiGiuong == doi) {*

*tongSoPhongGiuongDoi++;*

*}*

*temp = temp->next;*

*}*

*Tong so phong co giuong doi: << tongSoPhongGiuongDoi << endl;*

*}*

*// Hien thi danh sach phong co giuong doi*

*void hienThiPhongGiuongDoi() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Danh sach phong co giuong doi:*

*while (temp) {*

*if (temp->loaiGiuong == Doi || temp->loaiGiuong == doi) {*

*ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², so giuong: << temp->soGiuong << , Trang thai: << (temp->daDat ? Da dat : con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong giuong doi nao! << endl;*

*}*

*}*

*// Sap xep phong theo loai phong (tu A den Z)*

*void sapXepPhongTheoLoaiPhong() {*

*if (!dau || !dau->next) {*

*Danh sach phong trong hoặc chi co mot phong, khong can sap xep! << endl;*

*return;*

*}*

*bool swapped;*

*Phong\* temp;*

*Phong\* last = nullptr;*

*do {*

*swapped = false;*

*temp = dau;*

*while (temp->next != last) {*

*if (temp->loaiPhong > temp->next->loaiPhong) {*

*// Hoán đổi dữ liệu hai phong*

*swap(temp->masoPhong, temp->next->masoPhong);*

*swap(temp->loaiPhong, temp->next->loaiPhong);*

*swap(temp->giaPhong, temp->next->giaPhong);*

*swap(temp->dientich, temp->next->dientich);*

*swap(temp->soGiuong, temp->next->soGiuong);*

*swap(temp->loaiGiuong, temp->next->loaiGiuong);*

*swap(temp->daDat, temp->next->daDat);*

*swapped = true;*

*}*

*temp = temp->next;*

*}*

*last = temp;*

*} while (swapped);*

*Danh sach phong da duoc sap xep theo loai phong (tu A den Z)! << endl;*

*}*

*// Hien thi cac phong co loai phong cu the*

*void hienThiPhongTheoLoaiPhong(string loaiPhong) {*

*bool timThay = false;*

*Phong\* temp = dau;*

*Danh sach phong loai << loaiPhong << :*

*while (temp) {*

*if (temp->loaiPhong == loaiPhong) {*

*Ma so: << temp->masoPhong << , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², so giuong: << temp->soGiuong << , loai giuong: << temp->loaiGiuong*

*<< , Trang thai: << (temp->daDat ? Da dat : con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong loai << loaiPhong << ! << endl;*

*}*

*}*

*//Hien thi phong theo so giuong*

*void hienthiPhongTheoSoGiuong(int soGiuongCanTim) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Cac phong co << soGiuongCanTim << giuong:*

*while (temp) {*

*if (temp->soGiuong == soGiuongCanTim) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², loai giuong: << temp->loaiGiuong*

*<< , Trang thai: << (temp->daDat ? Da dat : con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong nao co << soGiuongCanTim << giuong! << endl;*

*}*

*}*

*//Hien thi phong theo loai giuong (Don/doi)*

*void hienthiPhongTheoLoaiGiuong(string loaiGiuongCanTim) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Cac phong co giuong loai << loaiGiuongCanTim << :*

*while (temp) {*

*if (temp->loaiGiuong == loaiGiuongCanTim) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², so giuong: << temp->soGiuong*

*<< , Trang thai: << (temp->daDat ? Da dat : con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong nao co giuong loai << loaiGiuongCanTim << ! << endl;*

*}*

*}*

*//Sap xep phong theo dien tich tang dan*

*void sapXepPhongTheoDienTichTangDan() {*

*if (!dau || !dau->next) {*

*Danh sach phong trong hoặc chi co mot phong, khong can sap xep! << endl;*

*return;*

*}*

*bool swapped;*

*Phong\* temp;*

*Phong\* last = nullptr;*

*do {*

*swapped = false;*

*temp = dau;*

*while (temp->next != last) {*

*if (temp->dientich > temp->next->dientich) {*

*// Hoan doi du lieu hai phong*

*swap(temp->masoPhong, temp->next->masoPhong);*

*swap(temp->loaiPhong, temp->next->loaiPhong);*

*swap(temp->giaPhong, temp->next->giaPhong);*

*swap(temp->dientich, temp->next->dientich);*

*swap(temp->soGiuong, temp->next->soGiuong);*

*swap(temp->loaiGiuong, temp->next->loaiGiuong);*

*swap(temp->daDat, temp->next->daDat);*

*swapped = true;*

*}*

*temp = temp->next;*

*}*

*last = temp;*

*} while (swapped);*

*Danh sach phong da duoc sap xep theo dien tich tang dan! << endl;*

*}*

*//Thong ke phong theo loai giuong (Don/doi)*

*void thongKePhongTheoLoaiGiuong() {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*int soPhongDon = 0, soPhongDoi = 0;*

*Phong\* temp = dau;*

*while (temp) {*

*if (temp->loaiGiuong == Don || temp->loaiGiuong == don) {*

*soPhongDon++;*

*} else if (temp->loaiGiuong == Doi || temp->loaiGiuong == doi) {*

*soPhongDoi++;*

*}*

*temp = temp->next;*

*}*

*So phong giuong don: << soPhongDon << endl;*

*So phong giuong doi: << soPhongDoi << endl;*

*}*

*//Tim phong theo loai giuong (Don/doi)*

*void timPhongTheoLoaiGiuong(string loaiGiuong) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Danh sach phong giuong << loaiGiuong << :*

*while (temp) {*

*if (temp->loaiGiuong == loaiGiuong) {*

*ma so: << temp->masoPhong << , loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², so giuong: << temp->soGiuong*

*<< , Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong giuong << loaiGiuong << ! << endl;*

*}*

*}*

*//Cap nhat so giuong của phong*

*void suaSoGiuongPhong(int maSoPhong, int soGiuongMoi) {*

*Phong\* phong = timPhong(maSoPhong);*

*if (!phong) {*

*Khong tim thay phong co ma so << maSoPhong << ! << endl;*

*return;*

*}*

*if (soGiuongMoi <= 0) {*

*So giuong moi khong hop le! << endl;*

*return;*

*}*

*phong->soGiuong = soGiuongMoi;*

*So giuong phong << maSoPhong << da duoc cap nhat thanh cong! << endl;*

*}*

*void hienThiPhongTheoGiaNhoHonHoacBang(double giaMax) {*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*return;*

*}*

*bool timThay = false;*

*Phong\* temp = dau;*

*Danh sach phong co gia nho hon hoac bang << giaMax << VND:*

*while (temp) {*

*if (temp->giaPhong <= giaMax) {*

*Ma so: << temp->masoPhong << , Loai: << temp->loaiPhong*

*<< , gia: << temp->giaPhong << VND, Dien tich: << temp->dientich*

*<< m², so giuong: << temp->soGiuong*

*<< , Trang thai: << (temp->daDat ? Da dat : Con trong) << endl;*

*timThay = true;*

*}*

*temp = temp->next;*

*}*

*if (!timThay) {*

*Khong co phong nao co gia nho hon hoac bang << giaMax << VND! << endl;*

*}*

*}*

*// Hien thi menu*

*void menu() {*

*int chon, soPhong, maSoPhong;*

*string loaiPhong, trangThai;*

*double gia, giaMin, giaMax;*

*while (true) {*

*==== MENU QUAN LY PHONG =====*

*1. Them phong moi*

*2. Kiem tra phong*

*3. Hien thi danh sach phong*

*4. Dat phong*

*5. Chinh sua thong tin phong*

*6. Xoa phong*

*7. Luu danh sach phong vao file*

*8. Doc danh sach phong tu file*

*9. Tim kiem phong theo loai*

*10. Hien thi phong co so giuong va loai giuong cu the*

*11. Huy dat phong*

*12. Tim phong co gia cao nhat*

*13. Thong ke so luong phong*

*14. Hien thi danh sach phong da dat*

*15. Hien thi danh sach phong con trong*

*16. Tinh tong doanh thu tu cac phong da dat*

*17. Sap xep phong theo gia (tang dan)*

*18. Dat nhieu phong cung luc*

*19. Xoa toan bo danh sach phong*

*20. Tim phong theo ma so phong*

*21. Hien thi tong so luong phong*

*22. Sua gia phong (neu co)*

*23. Tim kiem phong theo khoang gia (Tu X den Y)*

*24. Tim kiem phong theo trang thai (Da dat/Con trong)*

*25. Hien thi danh sach phong duoi dang bang*

*26. Tim phong co dien tich lon nhat*

*27. Thong ke so luong giuong theo loai phong*

*28. Tinh tong so giuong trong tat ca cac phong*

*29. Hien thi so phong co giuong doi con trong*

*30. Tinh tong so phong co giuong doi*

*31. Hien thi danh sach phong co giuong doi*

*32. Sap xep phong theo loai phong (A-Z)*

*33. Hien thi phong theo loai phong*

*34. Sap xep phong theo dien tich (tang dan)*

*35. Hien thi phong theo so giuong*

*36. Hien thi phong theo gia nho hon hoac bang mot gia tri cu the*

*37. Thong ke so phong theo loai giuong (don/doi)*

*38. Tim phong theo loai giuong (Don/doi)*

*39. Sua so giuong cua phong*

*40. Thoat*

*Chon chuc nang: ;*

*cin >> chon;*

*switch (chon) {*

*case 1:*

*themPhong();*

*break;*

*case 2:*

*Nhap ma so phong can kiem tra: ; cin >> soPhong;*

*if (timPhong(soPhong)) {*

*Phong << soPhong << co trong danh sach! << endl;*

*} else {*

*Phong << soPhong << khong ton tai! << endl;*

*}*

*break;*

*case 3:*

*hienThiPhong();*

*break;*

*case 4:*

*Nhap ma so phong can dat: ; cin >> soPhong;*

*datPhong(soPhong);*

*break;*

*case 5:*

*Nhap ma so phong can chinh sua: ; cin >> soPhong;*

*chinhSuaPhong(soPhong);*

*break;*

*case 6:*

*Nhap ma so phong can xoa: ; cin >> soPhong;*

*xoaPhong(soPhong);*

*break;*

*case 7:*

*luuDanhSachPhong();*

*break;*

*case 8:*

*docDanhSachPhong();*

*break;*

*case 9:*

*cout<<Nhap loai phong can tim: ;*

*cin>>loaiPhong;*

*timPhongTheoLoai(loaiPhong);*

*break;*

*case 10: {*

*int soGiuong;*

*string loaiGiuong;*

*Nhap so giuong: ;*

*cin >> soGiuong;*

*Nhap loai giuong (don/doi): ;*

*cin.ignore();*

*getline(cin, loaiGiuong);*

*hienThiPhongTheoSoGiuongVaLoaiGiuong(soGiuong, loaiGiuong);*

*break;*

*}*

*case 11:*

*cout<<Nhap so phong can huy dat: ;*

*cin>>soPhong;*

*huyDatPhong(soPhong);*

*break;*

*case 12:*

*timPhongGiaCaoNhat();*

*break;*

*case 13:*

*thongKePhong();*

*break;*

*case 14:*

*hienThiPhongDaDat();*

*break;*

*case 15:*

*hienThiPhongConTrong();*

*break;*

*case 16:*

*tinhTongDoanhThu();*

*break;*

*case 17:*

*sapXepPhongTheoGiaTangDan();*

*hienThiPhong();*

*break;*

*case 18:*

*datNhieuPhong();*

*break;*

*case 19:*

*xoaToanBoDanhSachPhong();*

*break;*

*case 20:*

*cout<<Nhap ma so phong can tim: ;*

*cin>>maSoPhong;*

*if (timPhong(maSoPhong))*

*{*

*cout<<Phong << maSoPhong<<da ton tai!<<endl;*

*}*

*else*

*{*

*cout<<Khong tim thay phong co ma so trong danh sach <<maSoPhong<<endl;*

*}*

*timPhongTheoMaSo(maSoPhong);*

*break;*

*case 21:*

*Tong so luong phong: << demSoLuongPhong() << endl;*

*break;*

*case 22:*

*Nhap ma so phong can cap nhat gia: ;*

*cin >> maSoPhong;*

*Nhap gia moi: ;*

*cin >> gia;*

*suaGiaPhong(maSoPhong,gia);*

*break;*

*case 23:*

*Nhap khoang gia (min max): ; cin >> giaMin >> giaMax;*

*timPhongTheoGia(giaMin, giaMax);*

*break;*

*case 24:*

*if (!dau) {*

*Danh sach phong trong! << endl;*

*break; // Ket thuc case 24 neu danh sach trong*

*}*

*Nhap trang thai phong (Da dat/Con trong): ;*

*cin.ignore();*

*getline(cin, trangThai);*

*if (trangThai.empty()) {*

*Bạn chưa nhap trang thai phong! << endl;*

*break;*

*}*

*timPhongTheoTrangThai(trangThai);*

*break;*

*case 25:*

*hienThiPhongDangBang();*

*break;*

*case 26:*

*timPhongDienTichLonNhat();*

*break;*

*case 27:*

*demSoGiuongTheoLoaiPhong();*

*break;*

*case 28:*

*tinhTongSoGiuong();*

*break;*

*case 29:*

*hienThiPhongGiuongDoiConTrong();*

*break;*

*case 30:*

*tinhTongSoPhongGiuongDoi();*

*break;*

*case 31:*

*hienThiPhongGiuongDoi();*

*break;*

*case 32:*

*sapXepPhongTheoLoaiPhong();*

*hienThiPhong();*

*case 33: {*

*string loaiPhong;*

*Nhap loai phong can tim: ;*

*cin.ignore();*

*getline(cin, loaiPhong);*

*hienThiPhongTheoLoaiPhong(loaiPhong);*

*break;*

*}*

*case 34:*

*sapXepPhongTheoDienTichTangDan();*

*hienThiPhong();*

*break;*

*case 35:*

*Nhap so giuong can tim: ;*

*cin >> soPhong;*

*hienthiPhongTheoSoGiuong(soPhong);*

*break;*

*case 36:*

*double giaMax;*

*cout<< Nhap gia toi da: ;*

*cin >> giaMax;*

*hienThiPhongTheoGiaNhoHonHoacBang(giaMax);*

*break;*

*case 37:*

*thongKePhongTheoLoaiGiuong();*

*break;*

*case 38: {*

*string loaiGiuong;*

*Nhap loai giuong (Don/doi): ;*

*cin.ignore();*

*getline(cin, loaiGiuong);*

*timPhongTheoLoaiGiuong(loaiGiuong);*

*break;*

*}*

*case 39:*

*Nhap ma so phong can sua so giuong: ;*

*cin >> maSoPhong;*

*Nhap so giuong moi: ;*

*cin >> soPhong;*

*suaSoGiuongPhong(maSoPhong, soPhong);*

*break;*

*case 40:*

*cout<<Da thoat khoi chuong trinh!*

*return;*

*default:*

*Lua chon khong hop le! << endl;*

*}*

*}*

*}*

*// Hàm main*

*int main() {*

*menu();*

*return 0;*

*}*