ĐỀ BÀI CHI TIẾT — ỨNG DỤNG QUẢN LÝ CÔNG VIỆC (TODO) CHẠY CONSOLE

> Mục tiêu: Luyện tập **Collection** (`List<T>`), thao tác **phương thức tĩnh** (static methods), **LINQ cơ bản** (`FindIndex`, `FindAll`, `OrderBy`), **serialize JSON** và **xử lý nhập/xuất console**.

---

1) Mô tả bài toán

Xây dựng chương trình Console **TodoApp** quản lý danh sách công việc. Mỗi công việc có các thuộc tính tối thiểu:

- `maCongViec` *(int)* – mã duy nhất.
- `tenCongViec` *(string)* – tên/miêu tả ngắn.
- `trangThai` *(bool)* – `false` = chưa hoàn thành, `true` = đã hoàn thành.

> **Ghi chú:** Để bám sát mã khởi tạo cho sẵn, danh sách dùng kiểu `List<object>` với **anonymous type** và truy cập qua `dynamic`.

---

2) Dữ liệu khởi tạo

Khởi tạo trước một số công việc mẫu:

```
List<object> lstTodo = new List<object>()
{
    new { maCongViec = 1, tenCongViec = "Ăn sáng", trangThai = false },
    new { maCongViec = 2, tenCongViec = "Ăn trưa", trangThai = false }
};
```

Có thể in nhanh danh sách dưới dạng JSON để kiểm tra:

```
Console.WriteLine(JsonSerializer.Serialize(lstTodo));
```

---

3) Menu chức năng (hiển thị lặp cho đến khi thoát)

```
------------- Hãy chọn 1 trong những chức năng sau của TodoApp -----------------
1. Thêm công việc vào danh sách
2. Hiển thị danh sách công việc
3. Cập nhật thông tin công việc (tên, giữ nguyên trạng thái)
4. Xóa công việc theo **mã**
5. Tìm kiếm công việc theo **tên** (không phân biệt hoa/thường)
6. Đánh dấu (tick) hoàn thành công việc theo **mã**
7. Sắp xếp và hiển thị **các công việc CHƯA hoàn thành** theo thứ tự (tùy chọn:
theo tên hoặc theo mã)
9. Thoát
```

Khi nhập số không hợp lệ, yêu cầu nhập lại. Nhập `9` để thoát.

---

4) Yêu cầu chức năng chi tiết

4.1 Thêm công việc (1)

- Nhập `maCongViec` *(int)* và `tenCongViec` *(string)* từ bàn phím.
- `trangThai` mặc định `false`.
- **Ràng buộc:** `maCongViec` không được trùng. Nếu trùng, thông báo lỗi và không thêm.
- Sau khi thêm, quay lại menu.

4.2 Hiển thị danh sách (2)

- In theo định dạng: `maCongViec - tenCongViec - trangThai` cho **tất cả** phần tử trong `lstTodo`.
- Nếu danh sách trống, in thông báo phù hợp.

4.3 Cập nhật thông tin (3)

- Nhập **mã** cần cập nhật.
- Nếu tìm thấy: cho phép nhập **tên công việc mới**; **giữ nguyên** `trangThai` hiện tại.
- Nếu **không tìm thấy**, in thông báo.

4.4 Xóa công việc (4)

- Nhập **mã** cần xóa.
- Tìm chỉ số bằng `FindIndex((dynamic item) => item.maCongViec == ma)`.
- Nếu tìm thấy: `RemoveAt(index)` và in “Xóa thành công”. Nếu không: in “Không tìm thấy…”.

4.5 Tìm kiếm theo tên (5)

- Nhập **từ khóa**; chuyển tên công việc sang **lowercase** rồi `Contains(tuKhoaLower)`.

- In tất cả kết quả phù hợp. Nếu không có, in “Không tìm thấy công việc nào chứa …”.

4.6 Đánh dấu hoàn thành (6)

- Nhập **mã** cần đánh dấu.
- Nếu tìm thấy: tạo **anonymous object** mới với cùng `maCongViec`, cùng `tenCongViec`, đặt `trangThai = true`, gán lại vào `lstCongViec[index]`.
- Nếu không: in thông báo không tìm thấy.

4.7 Sắp xếp công việc chưa hoàn thành (7)

- Lọc các phần tử có `trangThai == false`.
- **Bắt buộc:** cho phép **chọn tiêu chí** sắp xếp:
- `a) Theo tên` (tăng dần) dùng `OrderBy((dynamic x) => x.tenCongViec)`
- `b) Theo mã` (tăng dần) dùng `OrderBy((dynamic x) => x.maCongViec)`
- In danh sách sau sắp xếp. Nếu không còn công việc chưa hoàn thành, in thông báo.

4.8 Thoát (9)

- Kết thúc vòng lặp và đóng ứng dụng.

---

5) Cấu trúc chương trình & chữ ký phương thức

Tạo lớp tĩnh `Method` chứa các hàm sau (tên **giữ nguyên**):

- `HienThiMenuChucNang()` – In menu.
- `ThemCongViecVaoDanhSach(ref List<object> lstCongViec)`
- `hienThiCongViec(List<object> lstCongViec)`
- `capNhatThongTin(ref List<object> lstCongViec)`
- `xoaCongViec(ref List<object> lstCongViec)`
- `timKiemCongViecTheoTen(List<object> lstCongViec)`
- `tickHoanThanhCongViec(ref List<object> lstCongViec)`
- **(Bổ sung)** `sapXepChuaHoanThanh(List<object> lstCongViec)` hoặc gộp logic vào case 7.

Vòng lặp điều khiển (Program.cs)

- Dùng vòng lặp vô hạn `while (true)`; hiển thị menu; đọc lựa chọn; `switch-case` đến các phương thức; nhập `9` thì `break`.

---

6) Yêu cầu kỹ thuật & gợi ý triển khai

- Dùng `List<object>` và truy cập `` để phù hợp mã mẫu.
- Tạo **anonymous type** khi cần cập nhật/phát sinh phần tử mới để gán lại vị trí cũ trong List.
- Dùng `System.Text.Json` để **serialize** danh sách khi cần kiểm tra (không bắt buộc in ra mỗi bước).

- Chuẩn hóa chuỗi khi tìm kiếm: `ToLower().Trim()` cho cả tên công việc và từ khóa.
- Kiểm tra nhập liệu: bắt lỗi khi `Convert.ToInt32` (sai định dạng) bằng `try-catch` hoặc `int.TryParse` (khuyến khích).

---

7) Ví dụ tương tác

```
> 1
Nhập vào mã công việc: 3
Nhập vào tên việc: Đi chợ
=> Thêm thành công

> 2
1 - Ăn sáng - False
2 - Ăn trưa - False
3 - Đi chợ - False

> 6
Nhập mã công việc đã hoàn thành: 2
=> Đánh dấu thành công

> 7
Chọn tiêu chí sắp xếp (a: tên, b: mã): a
(Hiển thị các công việc chưa hoàn thành đã sắp xếp theo tên)

> 5
Nhập vào tên công việc cần tìm: an
1 - Ăn sáng - False
2 - Ăn trưa - True

> 9
Tạm biệt!
```

---

8) Tiêu chí chấm/đánh giá

1. Đủ **menu 1–7, 9** và chạy đúng yêu cầu.
2. Không sập chương trình khi nhập sai (xử lý lỗi cơ bản).
3. Tìm kiếm không phân biệt hoa/thường, kết quả chính xác.
4. Cập nhật **chỉ tên**, giữ nguyên `trangThai` (trừ chức năng số 6).
5. Sắp xếp case 7 hoạt động đúng theo tiêu chí chọn.
6. Mã nguồn rõ ràng, chia hàm trong lớp `Method`.

---

9) Mở rộng (không bắt buộc)

- Thêm thuộc tính `moTa` *(string)*, `hanChot` *(DateTime)*.
- Lưu/đọc danh sách ra/ từ file JSON.
- Thêm bộ lọc hiển thị: tất cả/đã hoàn thành/chưa hoàn thành.
- Cho phép **bỏ đánh dấu hoàn thành**.
- Chống nhập trùng `tenCongViec` trong cùng trạng thái (quy ước nhóm).

---

10) Gợi ý kiểm thử nhanh bằng LINQ

- Thử `OrderBy` / `Reverse` trên `List<int>` và `List<string>` để ôn tập quy tắc sắp xếp.
- Thử `OrderBy((dynamic item) => item.tenCongViec)` và `OrderBy((dynamic item) => item.maCongViec)` với danh sách công việc mẫu.

Kết quả kỳ vọng: Ứng dụng console hoàn chỉnh, có khả năng thêm/xóa/cập nhật/tìm kiếm/tick hoàn thành/sắp xếp theo yêu cầu, vận dụng đúng các thao tác **Collection** và **Method** trong C#.