

Bài 1: Kiểm Tra và Thông Kê Mật Khẩu

- **Mô tả:** Viết chương trình cho phép người dùng nhập vào một **mật khẩu** (dưới dạng chuỗi). Chương trình sẽ kiểm tra tính hợp lệ của mật khẩu dựa trên các quy tắc sau và thống kê các loại ký tự.
 - **Thao tác:**
 - Kiểm tra và hiển thị mật khẩu có **hợp lệ** hay không. Mật khẩu được coi là hợp lệ nếu:
 - Độ dài tối thiểu là **8 ký tự**.
 - Phải chứa ít nhất **một chữ cái in hoa** (A-Z).
 - Phải chứa ít nhất **một chữ cái thường** (a-z).
 - Phải chứa ít nhất **một chữ số** (0-9).
 - Phải chứa ít nhất **một ký tự đặc biệt** (ví dụ: **!@#\$%^&***).
 - Đếm và hiển thị tổng số lượng **ký tự đặc biệt** có trong mật khẩu.
 - Nếu mật khẩu không hợp lệ, chỉ ra **tất cả các lỗi** nó mắc phải.
 - **Input:**
 - Dòng 1: Một chuỗi ký tự (mật khẩu).
 - **Output:**
 - Dòng 1: Trạng thái mật khẩu: "Hợp lệ" hoặc "Không hợp lệ".
 - Dòng 2: Nếu không hợp lệ, liệt kê các lỗi (ví dụ: "Lỗi: Độ dài < 8; Thiếu chữ in hoa").
 - Dòng 3: Tổng số ký tự đặc biệt.

Bài 2: Struct Cơ Bản

- **Mô tả:** Định nghĩa cấu trúc **SanPham** bao gồm các trường: **MaSP** (chuỗi), **TenSP** (chuỗi), **DonGia** (số thực), **SoLuong** (số nguyên).
- **Thao tác:**
 - Viết hàm **nhap_san_pham** để nhập thông tin cho một sản phẩm.
 - Viết hàm **tinh_thanh_tien** nhận vào một cấu trúc **SanPham** và trả về giá trị **Thành Tiền** (Thành Tiền = Đơn Giá x Số Lượng).
 - Viết hàm **hien_thi_san_pham** để hiển thị tất cả thông tin và Thành Tiền.
- **Input:**
 - Dòng 1: Mã sản phẩm.
 - Dòng 2: Tên sản phẩm.
 - Dòng 3: Đơn giá và Số lượng (cách nhau bởi dấu cách).

- **Output:**

- Dòng 1: Mã SP: [Mã] - Tên: [Tên]
 - Dòng 2: Đơn giá: [Giá] - Số lượng: [SL]
 - Dòng 3: Thành tiền: [Thành Tiền]
-

Bài 3: Mảng Cấu Trúc (Quản Lý Ngày Tháng)

- **Mô tả:** Định nghĩa cấu trúc **NgayThang** bao gồm các trường: **Ngay** (số nguyên), **Thang** (số nguyên), **Nam** (số nguyên).
 - **Thao tác:**
 - Khai báo một **mảng cấu trúc** có kiểu **NgayThang** với N phần tử (ví dụ: N=5).
 - Nhập dữ liệu cho N ngày tháng.
 - Viết hàm **kiem_tra_hop_le** để kiểm tra xem một ngày tháng có hợp lệ không (ví dụ: tháng phải từ 1-12, ngày phải phù hợp với từng tháng).
 - Hiển thị danh sách các ngày tháng **hợp lệ** đã nhập.
 - **Input:**
 - Dòng 1: Số lượng ngày tháng N
 - N dòng tiếp theo: Mỗi dòng là Ngày, Tháng, Năm (cách nhau bởi dấu cách).
 - **Output:**
 - Các ngày tháng hợp lệ (dưới định dạng Ngày/Tháng/Năm) trên các dòng khác nhau.
-

Bài 4: Struct Lồng Nhau và Con Trỏ

- **Mô tả:** Định nghĩa cấu trúc **DiaChi** gồm: **SoNha** (số nguyên), **TenDuong** (chuỗi), **QuanHuyen** (chuỗi). Sau đó, định nghĩa cấu trúc **NhanVien** gồm: **MaNV** (chuỗi), **TenNV** (chuỗi), và **DiaChiNha** (là một biến kiểu **DiaChi** - **struct lồng nhau**).
- **Thao tác:**
 - Khai báo một biến có kiểu **NhanVien**.
 - Hiển thị toàn bộ thông tin của nhân viên (bao gồm cả địa chỉ).
- **Input:**
 - Dòng 1: Mã NV, Tên NV (cách nhau bởi dấu cách).
 - Dòng 2: Số nhà, Tên đường (cách nhau bởi dấu cách).
 - Dòng 3: Quận/Huyện.
- **Output:**
 - Dòng 1: Mã NV: [Mã] - Tên: [Tên]
 - Dòng 2: Địa chỉ: [Số nhà] [Tên đường], [Quận/Huyện]

Bài 5: Mảng Cấu Trúc và Phân Loại (Kết Quả Học Tập)

- **Mô tả:** Sử dụng lại cấu trúc **SinhVien** (MãSV, Tên, ĐTB). Chương trình quản lý mảng sinh viên.
- **Thao tác:**
 - Viết hàm nhập N sinh viên vào mảng cấu trúc.
 - Viết hàm **phan_loai** nhận vào một cấu trúc **SinhVien** và trả về kết quả phân loại học lực (ví dụ: "Giỏi" nếu ĐTB \geq 8.0, "Khá" nếu $ĐTB \geq 6.5$, "Trung Bình" nếu $ĐTB \geq 5.0$, "Yếu" nếu $ĐTB < 5.0$).
 - Hiển thị danh sách sinh viên theo từng loại học lực (Hiển thị các sinh viên Giỏi trước, sau đó là Khá, v.v.).
- **Input:**
 - Dòng 1: Số lượng sinh viên N.
 - N khối dữ liệu tiếp theo (Mã SV, Tên, ĐTB).
- **Output:**
 - Danh sách sinh viên Giỏi (Mã SV - Tên - ĐTB - Phân loại)
 - Danh sách sinh viên Khá (Mã SV - Tên - ĐTB - Phân loại)
 - ...