Bài tập C# căn bản

# Thuật toán

## Bài 1: Chuyển đổi hệ cơ số

Viết chương trình:

* Chuyển đổi số từ hệ cơ số 10 sang hệ cơ số 2
* Chuyển đổi số từ hệ cơ số 16 sang hệ cơ số 2
* Chuyển đổi số từ hệ cơ số 8 sang hệ cơ số 2

## Bài 2: Chuẩn hóa chuỗi

Viết hàm chuẩn hóa chuỗi với các yêu cầu như sau:

* Các chữ cái bắt đầu của một từ phải viết hoa. Các chữ cái sau viết thường.
* Các từ trong chuỗi chỉ được cách nhau một dấu cách.

Ví dụ:

Đầu vào: “ NgUyeN TROng NhAN”

Đầu ra: “Nguyen Trong Nhan”

## Bài 3: Mảng một chiều

Viết chương trình:

* Nhập số phần tử và giá trị (nguyên) của các phần tử trong mảng từ bàn phím, hiển thị mảng vừa nhập.
* Tìm và hiển thị phần tử lớn nhất, nhỏ nhất trong mảng cùng vị trí của phần tử đó trong mảng.
* Tìm và hiển thị số nguyên lớn nhất, nhỏ nhất trong mảng cùng vị trí của số đó trong mảng.
* Sắp xếp mảng tăng dần và giảm dần theo giá trị của các phần tử.

## Bài 4: Mảng hai chiều (ma trận)

Viết chương trình:

* Nhập vào số hàng, số cột và các giá trị tương ứng với từng vị trí trong ma trận từ bàn phím, hiển thị ma trận vừa nhập.
* Sắp xếp và hiển thị ma trận với điều kiện: giá trị trong từng hàng tăng dần.
* Sắp xếp và hiển thị ma trận với điều kiện: giá trị trong từng cột giảm dần.

## Bài 5: Forever Alone Person

**Mô tả bài toán:**

Dù thông minh, đẹp trai, học giỏi nhưng vẫn không thoát khỏi kiếp FA vì chỉ có 8cm và lười tắm, Đạt 09 lại buồn bã trở về vnoi code lại từ đầu. Trong một ngày chán như con gián, Đạt 09 nhổ trộm bông hướng dương nhà hàng xóm và ngồi … đếm cánh hoa. Mỗi lần một cánh hoa rụng xuống là câu nói “Tắm”, “Không tắm” lại vang lên. Đã ba năm trôi qua Đạt 09 vẫn chỉ ngồi đếm lá và chưa tắm rồi đột nhiên anh ta đứng lên và chạy về máy tính: “Đúng rồi số ngẫu nhiên!”. Hóa ra Đạt 09 đã nghĩ ra cách làm mới mà không phải nhổ trộm hoa + ngồi đếm số cánh hoa. Chúng ta biết rằng số ngẫu nhiên được sinh ra bởi bộ ba số a, b, m theo quy tắc:

* Số thứ nhất là x1=b mod m
* Số thứ k là (a\*xk-1+b) mod m với k>1

**Đầu vào:**

* Dòng 1: Số nguyên T (1≤T≤10) là số lượng test
* T dòng tiếp theo, mỗi dòng gồm 4 số nguyên a, b, m, k (1≤a, b, m, k≤1015)

**Đầu ra:**

Gồm T dòng, mỗi dòng in ra số xk tương ứng.

**Ví dụ:**

*Đầu vào:*

T = 3

1 1 1 1

2 5 100 6

1 8 777 6

*Đầu ra:*

0

15

48

Yêu cầu:

* Nhập số T từ bàn phím
* Nhập giá trị a, b, m, k trong T dòng tiếp theo từ bàn phím, mỗi giá trị cách nhau bởi 1 dấu cách

# OOP

## Bài 1: Phân số

Xây dựng lớp **PhanSo** để biểu diễn khái niệm phân số với hai thành phần dữ liệu **tử số**, **mẫu số** và các hàm thành phần **cộng**, **trừ**, **nhân**, **chia** hai phân số, các hàm thành phần **xuất**, **nhập**, **định giá trị** cho phân số. Viết chương trình cho phép nhập vào hai phân số, in ra kết quả các phép toán cộng, trừ, nhân, chia hai phân số kể trên.

## Bài 2: Số phức

Xây dựng lớp biểu diễn khái niệm **số phức** với hai thành phần dữ liệu **thực**, **ảo** và các hàm thành phần **xuất**, **nhập**, **định giá trị** cho **số phức** và các phép toán **cộng**, **trừ**, **nhân**, **chia** hai số phức. Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên.

## Bài 3: Kế thừa

**Phần 1:**

Tạo lớp người chứa thông tin

* Tên
* Giới tính
* Ngày sinh
* Địa chỉ

Với hàm khởi tạo không tham số và đầy đủ tham số.

1. Viết phương thức InputInfo(), nhập thông tin người từ bàn phím
2. Viết phương thức ShowInfo(), hiển thị tất cả thông tin người

**Phần 2:**

Tạo lớp sinh viên kế thừa từ người, lưu trữ thông tin một sinh viên

* Mã sinh viên
* Điểm trung bình
* Email

1. Override phương thức InputInfo(), nhập thông tin sinh viên từ bàn phím
2. Override phương thức ShowInfo(), hiển thị tất cả thông tin sinh vien
3. Viết phương thứ xét xem sinh viên có được học bổng hay không? (điểm trung bình trên 8.5)

**Phần 3:**

Chỉnh sửa hàm main để chạy chức năng của lớp sinh viên

Tạo menu chọn như sau:

* Chọn 1: Nhập vào n sinh viên (n là số lượng sinh viên, được nhập từ bàn phím)
* Chọn 2: Hiển thị thông tin tất cả sinh viên ra màn hình
* Chọn 3: Hiển thị sinh viên có điểm trung bình cao nhất và điểm trung bình thấp nhất
* Chọn 4: Tìm kiếm sinh viên theo mã sinh viên. Nhập vào mã sinh viên, nếu tồn tại sinh viên có mã đó thì in thông tin sinh viên ra màn hinh, nếu không tồn tại thì in “Không có sinh viên với mã sinh viên là {giá trị của mã sinh viên}”
* Chọn 5: Hiển thị danh sách các sinh viên được học bổng, sắp xếp theo thứ tự điểm từ cao xuống thấp
* Chọn 6: Thoát

**Phần 4:**

Tạo lớp giao viên kế thừa từ lớp người, lưu trữ thông tin giao viên như:

* Mã lớp dạy
* Lương một giờ
* Số giờ dậy trong tháng
* Thâm niên

1. Override phương thức InputInfo(), nhập thông tin giao viên từ bàn phím
2. Override phương thức ShowInfo(), hiển thị thông tin giao viên
3. Viết phương thức tính lương thực nhận, trả về lương thực nhận theo công thức:
   * Nếu (thâm niên < 1) thì Lương thực nhận = (lương một giờ \* số giờ dạy trong tháng)
   * Nếu (thâm niên < 3) thì Lương thực nhận = (lương một giờ \* số giờ dạy trong tháng) \* 1.1
   * Nếu (thâm niên < 5) thì Lương thực nhận = (lương một giờ \* số giờ dạy trong tháng) \* 1.3
   * Nếu (thâm niên >= 5) thì Lương thực nhận = (lương một giờ \* số giờ dạy trong tháng) \* 1.5

**Phần 5:**

Chỉnh sửa hàm main để chạy các chức năng của lớp giao viên

Tạo menu lựa chọn như sau:

* Chọn 1: Nhập vào n giảng viên (n là số lượng giảng viên, được nhập từ bàn phím)
* Chọn 2: Hiển thị thông tin tất cả giảng viên ra màn hình
* Chọn 3: Hiển thị giảng viên có lương cao nhất
* Chọn 4: Thoát.

## Bài 4: Interface

**Phần 1:**

Xây dựng lớp “DaySo” để mô tả một dãy số bao gồm các phương thức sau:

* Nhap() dùng để nhập dãy số từ bàn phím
* Print() dùng để in dãy số ra màn hình
* Hàm tạo DaySo(int n) dùng để khởi tạo một mảng gồm n phần tử

**Phần 2:**

Xây dựng interface Sort với phương thức Sort().

**Phần 3:**

Xây dựng lớp QuickSort, SelectionSort, InsertSort bằng cách kế thừa từ lớp DaySo và triển trai interface Sort để thực hiện các thuật toán sắp xếp tương ứng.

## Bài 5: Abstract class

**Phần 1:**

Tạo một lớp trừu tượng có tên Phone chứa những phương thức trừu tượng sau đây:

* Public abstract void InsertPhone(string name, string phone)
* Public abstract void UpdatePhone(string name, string newPhone)
* Public abstract void DeletePhone(string name)
* Public abstract void SearchPhone(string name)

**Phần 2:**

Tạo lớp có tên PhoneBook kế thừa lớp Phone:

* Tạo một List để lưu dữ liệu
* Phương thức InsertPhone:
  + Nếu tên (name) chưa tồn tại thì thêm mới số điện thoại (phone) tương ứng vào List
  + Nếu tên (name) đã tồn tại thì ghi thêm số điện thoại (phone) tương ứng vào List
* Phương thức UpdatePhone:
  + Nếu tên (name) đã tồn tại thì thay thế số điện thoại cũ bằng số điện thoại mới (newphone)
  + Nếu tên (name) không tồn tại thì thông báo tên không tồn tại
* Phương thức DeletePhone:
  + Nếu tên (name) đã tồn tại thì xóa bỏ khỏi List
  + Nếu tên (name) không tồn tại thì thông báo tên không tồn tại
* Phương thức SearchPhone: Tìm kiếm số điện thoại theo tên (name)

**Phần 3:**

Chỉnh sửa hàm main để chạy các chức năng của lớp PhoneBook

Tạo menu lựa chọn như sau:

* Chọn 1: Thêm mới Phone vào List
* Chọn 2: Chỉnh sửa Phone trong List
* Chọn 3: Xóa Phone khỏi List
* Chọn 4: Tìm kiếm theo tên
* Chọn 5: Thoát