



Bài tập thực hành

Ghi nhớ:

1. Struct khi khai báo kết thúc bằng dấu chấm phẩy (;).
2. Muốn khởi tạo một biến có sử dụng cấu trúc ngay khi khai báo cần đặt trong cặp {}, phải truyền đúng trật tự và đủ dữ liệu như đã khai báo bên trong struct.
3. Để truy cập đến các thành phần trong struct, sử dụng toán tử chấm (.).
4. Có thể sử dụng mảng để quản lý tập hợp các biến kiểu struct.

2.1. Bài tập 1

1. **Thời lượng:** 10 phút.
2. **Mô tả bài toán:** Xây dựng một cấu trúc để quản lý thông tin của một nhân viên bao gồm: Mã (tối đa 5 ký tự), họ tên (tối đa 50 ký tự), chức vụ (tối đa 20 ký tự), thâm niên công tác (số nguyên), số ngày phép (số nguyên).

- **Yêu cầu 1:** Khai báo và khởi tạo 5 nhân viên với các thông tin sau:

Mã NV	Họ tên	Chức vụ	TNCT	Phép
DI01	Nguyễn Kim Long	Giám đốc	47	17
AC05	Đậu Thị Duyên	Kế toán	47	25
HR03	Trần Hạ Lan	Nhân sự	22	7
TR02	Trần Ngọc Thoa	Giáo vụ	13	9
IT04	Trần Ngọc Đăng	IT	4	2

- **Yêu cầu 2:** Xuất thông tin 5 nhân viên ra màn hình.

2.2. Bài tập 2

1. **Thời lượng:** 20 phút.
2. **Mô tả bài toán:** Xây dựng một cấu trúc để quản lý thông tin của quyển sách bao gồm: Tựa đề (tối đa 100 ký tự), tác giả (tối đa 50 ký tự), số trang (số nguyên), giá tiền (số nguyên).

- **Yêu cầu 1:** Khai báo 1 mảng sách tối đa 50 cuốn. Yêu cầu người dùng nhập vào số lượng sách cần lưu trữ.
- **Yêu cầu 2:** Yêu cầu người dùng nhập vào thông tin từng cuốn sách để lưu trữ.
- **Yêu cầu 3:** Xuất danh sách tất cả thông tin sách.
- **Yêu cầu 4:** Tìm kiếm và hiển thị những cuốn sách có từ 200 trang trở lên.
- **Yêu cầu 5:** Tìm kiếm và hiển thị những cuốn sách có giá từ 90000 trở xuống.

2.3. Bài tập 3

1. **Thời lượng:** 25 phút.
2. **Mô tả bài toán:** Xây dựng một cấu trúc để quản lý thông tin của sinh viên bao gồm: mã số SV (tối đa 6 ký tự), họ tên (tối đa 50 ký tự), lớp (tối đa 6 ký tự), điểm toán, điểm anh văn, điểm tin học, điểm trung bình và xếp loại (tối đa 15 ký tự).

- **Yêu cầu 1:** Khai báo 1 mảng sinh viên tối đa 100 người. Yêu cầu người dùng nhập vào số lượng sinh viên cần lưu trữ.
- **Yêu cầu 2:** Tính điểm trung bình của từng sinh viên và xếp loại như sau:
 - **Giỏi:** $TB \geq 8.0$
 - **Khá:** $6.5 \leq TB < 8.0$
 - **Trung bình:** $5.0 \leq TB < 6.5$
 - **Yếu:** $3.5 \leq TB < 5.0$
 - **Kém:** $TB < 3.5$
- **Yêu cầu 3:** Xuất danh sách sinh viên vừa nhập theo cấu trúc dưới đây.

MSSV	Họ tên	Lớp	Toán	AV	Tin	ĐTB	Xếp loại
213001	Nguyễn Văn A	QL131	8.0	8.8	6.8	7.9	Khá
213002	Nguyễn Văn B	QL141	4.0	5.0	7.0	5.3	Trung bình
213003	Nguyễn Thị C	QL141	6.0	2.0	4.0	4.0	Yếu

- **Yêu cầu 4:** Đếm số lượng sinh viên học lực Giỏi, Khá, TB, Yếu và Kém (Nếu không có ghi là 0).
- **Yêu cầu 5:** Tìm thông tin sinh viên bằng cách nhập vào MSSV.

2.4. Bài tập 4

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Xây dựng bảng xếp hạng các bài hát, biết rằng cấu trúc của một bài hát bao gồm: Mã bài hát (tối đa 6 ký tự), tên bài hát (tối đa 100 ký tự), lượt nghe (số nguyên).
 - **Yêu cầu 1:** Tìm ra ca khúc HIT (ca khúc có nhiều lượt nghe nhất)
 - **Yêu cầu 2:** Hiển thị danh sách xếp hạng dựa vào số lượt nghe giảm dần.

2.5. Bài tập 5

1. **Thời lượng:** 30 phút.
 2. **Mô tả bài toán:** Xây dựng một cấu trúc để quản lý thông tin một mẫu giày bao gồm:
 - **ID:** Mã giày – (chuỗi tối đa 5 ký tự)
 - **Size:** Kích cỡ giày – (số thực. VD: 7.0, 7.5, 8.5, ...)
 - **GiớiTính:** Giới tính – (0. Dành cho nữ, 1. Dành cho nam, 2. Unisex)
 - **SoLuong:** Số lượng
 - **DonGia:** Đơn giá
- a) Khai báo 1 biến mảng toàn cục có tối đa 50 phần tử giày và biến toàn cục số lượng.
- b) Xây dựng chương trình con:
- Hàm **XuatGiay**: Xuất thông tin giày.
 - Hàm **NhapGiay**: Nhập thông tin giày theo số lượng giày của người nhập.
 - Hàm **TimGiay**: Tìm giày dựa vào giới tính và kích thước của người nhập.
 - Hàm **MuaGiay**: Yêu cầu người dùng nhập vào mã giày và số lượng muốn mua. Tìm kiếm mã giày phù hợp và kiểm tra số lượng giày có đủ đáp ứng số lượng mua không rồi tính tiền.

2.6. Bài tập 6

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Xây dựng một cấu trúc để quản lý thông tin thực phẩm bao gồm:

- Tên hàng hóa (tối đa 50 ký tự),
- Mã hàng (tối đa 6 ký tự).
- Số lượng.
- Đơn giá.

Sau khi nhập sản phẩm. Tạo Menu thực hiện các chức năng sau:

- **Chức năng 1:** Xem thông tin các sản phẩm đang có.
- **Chức năng 2:** Tìm sản phẩm dựa vào mã hàng.
- **Chức năng 3:** Mua hàng.
- Sau khi thực hiện xong mỗi chức năng, yêu cầu khách hàng có muốn tiếp tục hay không? (Bấm Y để tiếp tục, Bấm N thì thoát khỏi chương trình)

Xây dựng chức năng:

- **Chức năng 1:** Hiển thị các sản phẩm vừa nhập theo bảng bên dưới

Tên hàng hóa	Mã hàng	Số lượng	Đơn giá
Coca cola	N-001	20	4.800
Pepsi	N-002	25	4.600
Hamburger	A-001	15	12.000

- **Chức năng 2:** Tìm kiếm sản phẩm dựa vào mã hàng (dùng strcmp để so sánh), nếu không có xuất thông báo "Không tìm thấy".
- **Chức năng 3:** Yêu cầu người dùng nhập vào mã hàng, số lượng mua. Nếu số lượng mua > số lượng có thì xuất thông báo "Không đủ hàng".

2.7. Bài tập 7

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Xây dựng cấu trúc của một phân số bao gồm: Tử số (số nguyên), mẫu số (số nguyên).

- **Yêu cầu 1:** Xây dựng hàm **NhapPhanSo()** yêu cầu người dùng nhập tử số và mẫu số.
- **Yêu cầu 2:** Xây dựng hàm **XuatPhanSo()** truyền vào tham số phân số, xuất phân số này ra màn hình.
- **Yêu cầu 3:** Xây dựng hàm **Tong2PS()** truyền vào 2 phân số. Thực hiện tính tổng và trả về kết quả là kiểu phân số.
- **Yêu cầu 4:** Xây dựng hàm **Hieu2PS()** truyền vào 2 phân số. Thực hiện tính hiệu và trả về kết quả là kiểu phân số.
- **Yêu cầu 5:** Xây dựng hàm **RutGonPS()** truyền vào 1 phân số. Tiến hành rút gọn phân số về tối giản. Khi đã tối giản thì trả kết quả về kiểu phân số.
- **Yêu cầu 6:** Xây dựng hàm **KiemTraPS()** truyền vào tham số phân số. Kiểm tra phân số đưa vào là âm hay dương. Hàm trả về 1 khi phân số có giá trị dương, hàm trả về -1 khi phân số có giá trị âm.

2.8. Bài tập 8

1. **Thời lượng:** 45 phút.
2. **Mô tả bài toán:** Xây dựng chương trình mô phỏng chuyển khoản ngân hàng với thông tin tài khoản ngân hàng bao gồm: Mã tài khoản (chuỗi 12 chữ số), tên tài khoản (chuỗi tối đa 50 ký tự), Số dư (số nguyên).
 - a) Khai báo mảng toàn cục có tối đa 10 tài khoản.
 - b) Xây dựng chương trình con:
 - Hàm **TaoTaiKhoan** yêu cầu người dùng nhập các thông tin tài khoản, và lưu vào mảng tài khoản ngân hàng.
 - Hàm **TraCuuTK** truyền vào tham số mã tài khoản. Hàm tìm kiếm mã tài khoản có tồn tại trong mảng hay không? Nếu có, hàm trả về trị số tìm thấy trong mảng. Ngược lại, hàm trả về giá trị -1.
 - Hàm **TimTK** yêu cầu người dùng nhập vào mã tài khoản cần tìm, gọi hàm **TraCuuTK** để tìm kiếm tài khoản dựa vào mã tài khoản vừa nhập. Khi hàm **TraCuuTK** cho kết quả có giá trị khác -1, hàm hiển thị thông tin tài khoản ngân hàng và in số dư ra màn hình, ngược lại hiển thị thông báo “*Tài khoản không tìm thấy*”. Hàm không có giá trị trả về.
 - Hàm **ChuyenKhoan** gọi hàm **TimTK** để truy xuất thông tin tài khoản chuyển và nhận tiền. Nếu cả 2 tài khoản đều được tìm thấy thì yêu cầu người dùng nhập vào số tiền muốn chuyển. Thực hiện chuyển tiền và thay đổi trực tiếp số dư của mỗi tài khoản chuyển và nhận. Sau khi chuyển tiền thành công, lưu biến giao dịch vào mảng giao dịch.
 - c) Tại chương trình chính, xây dựng menu chức năng. Mỗi chức năng sau khi thực hiện xong tự động quay về menu chức năng. Chương trình gồm các chức năng:
 - 1. Đăng ký tài khoản ngân hàng.
 - 2. Xem thông tin số dư tài khoản.
 - 3. Chuyển khoản.

2.9. Bài tập 9

1. **Thời lượng:** 60 phút.
2. **Mô tả bài toán:** Xây dựng chương trình quản lý phòng NET gồm tập hợp 12 máy tính. Mỗi máy tính có cấu trúc gồm: Mã máy (số nguyên), Giờ vào (mảng số nguyên có 2 phần tử), Giờ ra (mảng số nguyên có 2 phần tử), Trạng thái (số nguyên – 1: máy bận, 0: máy rảnh).
 - a) Khai báo mảng toàn cục. Sử dụng vòng lặp để gán trực tiếp mã máy theo thứ tự từ 1-12, giờ vào/giờ ra đều set là 0, tất cả các máy đều ở trạng thái rảnh.
 - b) Tại chương trình chính, xây dựng menu chức năng. Mỗi chức năng sau khi thực hiện xong tự động quay về menu chức năng. Chương trình gồm các chức năng:

<ul style="list-style-type: none">▪ 1. Mở máy▪ 2. Khoá máy	<ul style="list-style-type: none">▪ 3. Hiển thị danh sách phòng máy▪ 4. Tìm kiếm máy rảnh
---	--

c) Xây dựng chương trình con:

- Hàm **KiemTraTrangThai** truyền tham số mã máy, hàm kiểm tra trạng thái máy và trả về kết quả là 1 nếu máy bận, là 0 nếu máy rảnh.
- Hàm **MoKhoaMay** truyền tham số mã máy. Gọi hàm **KiemTraTrangThai**, nếu hàm trả về 0 thì yêu cầu nhập giờ vào, thay đổi trạng thái từ rảnh sang bận. Ngược lại, thông báo “*Máy hiện đã có khách. Vui lòng chọn máy khác.*”
- Hàm **TinhTien** truyền tham số mã máy. Hàm trả về thời lượng chơi game dựa vào giờ vào và giờ ra. Biết rằng 1 giờ thu được 10.000 VNĐ.
- Hàm **KhoaMay** truyền tham số mã máy. Gọi hàm **KiemTraTrangThai**, nếu hàm trả về 1 thì yêu cầu nhập giờ ra, gọi hàm **TinhTien**, hiển thị số tiền. Sau khi tính tiền, reset giờ vào và giờ ra về 0, trạng thái máy quay về rảnh. Ngược lại, hiển thị thông báo “*Máy hiện không có khách, không thể khoá máy.*”
- Hàm **TimMayRanh** hiển thị thông tin các máy có trạng thái “*Rảnh*”.
- Hàm **HienThiDS** hiển thị toàn bộ thông tin các máy theo cấu trúc sau:

Mã máy	Giờ vào	Giờ ra	Trạng thái
PC01	00:00	00:00	Rảnh
PC02	10:00	00:00	Bận

2.10. Bài tập 10

1. Thời lượng: 60 phút.

2. Mô tả bài toán: Xây dựng 1 cấu trúc **PHONGHOC** gồm các thông tin:

- **MaPH**: Mã phòng học – (chuỗi 5 ký tự)
- **LoaiPH**: Loại phòng học – (chuỗi 2 ký tự - LT: Lý thuyết. TH: Thực hành)
- **SL**: Số lượng chỗ ngồi – (số nguyên dương)
- **TT**: Tình trạng (kiểu bool/int – 0: Available. 1: Busy)

a) Khai báo 1 biến mảng toàn cục có tối đa 50 phần tử **PHONGHOC** và biến toàn cục **SoluongPH = 0**.

b) Xây dựng chương trình con

- Hàm **int KiemTraMaPhong(char maphong[])**: Kiểm tra mã phòng đã tồn tại trong mảng các phòng học hay không? Nếu có hàm trả về giá trị 1, ngược lại, hàm trả về giá trị 0.
- Hàm **void TaoPhongHoc(PHONGHOC &ph)**: Nhập các thông tin một phòng học.
 - Mã phòng học là duy nhất cho mỗi phòng học nên sau khi nhập mã phòng học, gọi hàm **KiemTraMaPhong** để kiểm tra tính hợp lệ của mã phòng. Nếu mã phòng chưa tồn tại trong mảng thì chấp nhận, nếu đã tồn tại, thì yêu cầu nhập lại mã phòng mới.
 - Kiểm tra tính hợp lệ của loại phòng (chỉ chấp nhận “LT” hoặc “TH”), nếu không hợp lệ thì yêu cầu nhập lại.
 - Kiểm tra tính hợp lệ của số lượng chỗ ngồi (chỉ chấp nhận số ≥ 0), nếu không hợp lệ thì yêu cầu nhập lại.

- Hàm **void TaoCacPhongHoc()**: Yêu cầu người dùng nhập số lượng phòng muốn tạo. Dùng biến toàn cục **SoluongPH** để lưu lại giá trị nhập. Sử dụng vòng lặp và gọi hàm **TaoPhongHoc** để tạo từng phòng học.
- Hàm **void HienThiPhongHoc(PHONGHOC ph)**: Hiển thị thông tin của một phòng học dựa vào tham số **ph**. Xuất thông tin theo hàng ngang bên dưới:

P-503	Thực hành	12	Bận
-------	-----------	----	-----

- Hàm **void DSPhongHoc()**: Hiển thị thông tin của tất cả phòng học hiện có trong mảng theo nội dung sau:

=====

DANH SÁCH PHÒNG HỌC

=====

Mã phòng	Loại phòng	Số lượng	Tình trạng
----------	------------	----------	------------

Sử dụng vòng lặp và gọi hàm **HienThiPhongHoc** để hiển thị nội dung từng phòng học.

- Hàm **void DSPhongTheoSL(int soluong)**: Tìm phòng theo tiêu chí số lượng \geq tham số truyền vào. Nếu thoả điều kiện thì gọi hàm **HienThiPhongHoc** để xuất thông tin phòng học đó. Hãy đặt tiêu đề *"DANH SÁCH PHÒNG HỌC CÓ SỐ LƯỢNG \geq XX CHỖ"* trước khi hiển thị danh sách.
- Hàm **void DSPhongTheoTT(int tt)**: Tìm phòng theo tiêu chí tình trạng Bận hoặc Rảnh. Nếu thoả điều kiện thì gọi hàm **HienThiPhongHoc** để xuất thông tin phòng học đó. Hãy đặt tiêu đề *"DANH SÁCH PHÒNG HỌC CÓ TÌNH TRẠNG YY"* trước khi hiển thị danh sách.
- Hàm **void DSPhongTheoLoai(char loai[])**: Tìm phòng theo tiêu chí loại phòng Lý thuyết hoặc Thực hành. Nếu thoả điều kiện thì gọi hàm **HienThiPhongHoc** để xuất thông tin phòng học đó. Hãy đặt tiêu đề *"DANH SÁCH PHÒNG HỌC ZZ"* trước khi hiển thị danh sách.
- Hàm **int KiemTraTinhTrangConPhong(char maphong[])**: Kiểm tra tình trạng phòng truyền tham số mã phòng có đang bận hay không. Nếu có, hàm trả về giá trị 1, ngược lại, hàm trả về giá trị 0.
- Hàm **void DatPhong()**: Đặt phòng học.
 - Yêu cầu người dùng nhập vào mã phòng học.
 - Gọi hàm **KiemTraMaPhong** để kiểm tra nội dung nhập vào có tồn tại trong mảng hay không. Nếu không, hiển thị *"Mã phòng không tồn tại."* và yêu cầu nhập lại.
 - Gọi hàm **KiemTraTinhTrangConPhong** để kiểm tra phòng học đó có đang bận hay không. Nếu bận, hãy xuất thông tin *"Phòng học này đang được sử dụng. Vui lòng chọn phòng khác"*. Ngược lại, thì thay đổi tình trạng của phòng học đó từ Rảnh sang Bận, đồng thời xuất thông tin *"Bạn đã đặt phòng thành công."*

c) Tại chương trình chính, xây dựng menu chức năng. Mỗi chức năng sau khi thực hiện xong tự động quay về menu chức năng. Chương trình gồm các chức năng:

- Chức năng 1: Xem danh sách phòng học
- Chức năng 2: Tạo phòng học
- Chức năng 3: Tìm phòng học
- Chức năng 4: Đặt phòng

Xây dựng chức năng

- Chức năng 1: Gọi hàm **DSPhonghoc** để hiển thị toàn bộ phòng học.
- Chức năng 2: Gọi hàm **TaoCacPhongHoc**
- Chức năng 3: Tìm theo 3 tiêu chí:
 - Theo loại: Yêu cầu người dùng nhập loại phòng. Kiểm tra tính hợp lệ của loại phòng (chỉ chấp nhận "LT" hoặc "TH"), nếu không hợp lệ thì yêu cầu nhập lại. Ngược lại, gọi hàm **DSPhongTheoLoai**.
 - Theo số lượng: Yêu cầu người dùng nhập số lượng. Kiểm tra tính hợp lệ của số lượng (chỉ chấp nhận số nguyên dương), nếu không hợp lệ thì yêu cầu nhập lại. Ngược lại, gọi hàm **DSPhongTheoSL**.
 - Theo tình trạng: Yêu cầu người dùng nhập tình trạng. Kiểm tra tính hợp lệ của loại phòng (chỉ chấp nhận 1 hoặc 0), nếu không hợp lệ thì yêu cầu nhập lại. Ngược lại, gọi hàm **DSPhongTheoTT**.
- Chức năng 4: Gọi hàm **DatPhong**.



Câu hỏi ôn tập

(Xem đáp án ở trang)

3.1. Từ khoá nào sau đây được sử dụng để định nghĩa cấu trúc?

- | | |
|------------------|-----------------|
| A. struct | C. def_struct |
| B. structure_def | D. Cả 3 đều sai |

3.2. Các thành phần bên trong một kiểu cấu trúc được biết như là?

- | | |
|------------|-----------------|
| A. data | C. objects |
| B. members | D. Cả 3 đều sai |

3.3. Cho các phát biểu sau, hãy nhận xét phát biểu nào là đúng:

1. Chúng ta có thể khai báo 1 kiểu cấu trúc nằm lồng trong 1 cấu trúc khác.
2. Chúng ta không được bỏ từ khoá typedef khi khai báo cấu trúc.
3. Một kiểu cấu trúc không thể chứa nhiều thành phần có kiểu dữ liệu giống nhau.
4. Các biến sử dụng kiểu cấu trúc đều có thể là 1 tham số khi truyền vào hàm.

- | | |
|-------------------------------|----------------------------|
| A. Các phát biểu 2, 3, 4 đúng | C. Chỉ phát biểu 2, 3 đúng |
| B. Các phát biểu 1, 3, 4 đúng | D. Chỉ phát biểu 1, 4 đúng |

3.4. Ký hiệu nào được sử dụng để kết thúc khai báo một cấu trúc?

- | | |
|------|------|
| A. : | C. ; |
| B. } | D. , |

3.5. Hãy cho biết cách định nghĩa kiểu cấu trúc nào là đúng cú pháp?

- A. struct {int a;}
- B. struct a_struct{int a;}
- C. struct a_struct int a;
- D. struct a_struct {int a};

3.6. Cho đoạn code sau, hãy cho biết điều gì ngăn cản chương trình không thể compile?

```
1 struct Employee
2 {
3     int id;
4     float wage;
5 }
```

- A. Dòng 1: Thiếu () sau Employee.
- B. Dòng 1: Thiếu từ khoá typedef trước struct.
- C. Dòng 4: wage phải cùng kiểu dữ liệu với id.
- D. Dòng 5: Thiếu dấu ; sau ngoặc nhọn.

3.7. Cho đoạn code sau. Hãy cho biết lệnh nào để gán giá trị vào thành phần wage của emps thứ 2 trong mảng ?

```
struct Employee
{
    int id;
    float wage;
}

Employee emps[10];
```

- A. emps[2]->wage = 50.00;
- B. emps2.wage = 7.50;
- C. wage[2].emps = 29.75;
- D. emps[2].wage = 75.00;

3.8. Hãy cho biết kết quả của đoạn code sau:

```
#include <stdio.h>
void main()
{
    struct ShoeType
    {
        char style[10];
        double price;
    };
    ShoeType shoe1 = {"Addidas", 9.99};
    ShoeType shoe2 = shoe1;
    shoe2.price = shoe2.price / 3;
    printf("%s $%f %s $%f", shoe1.style, shoe1.price, shoe2.style, shoe2.price);
}
```

- A. Addidas \$9.99 Addidas \$9.99
- B. Addidas \$3.33 Addidas \$3.33
- C. Addidas \$9.99 Addidas \$3.33
- D. Addidas \$3.33 Addidas \$9.99

3.9. Cho đoạn code sau. Hãy cho biết cách xuất giá trị của thành phần gpa thuộc phần tử đầu tiên trong mảng?

```
struct Student
{
    int id;
    float gpa;
}
Student stud[10];
```

- A. printf("%f", stud[1].gpa);
- B. printf("%f", firststud.gpa);
- C. printf("%f", stud[0].gpa);
- D. printf("%f", stud0.gpa);

3.10. Hãy cho biết kết quả của đoạn code sau?

```
#include <stdio.h>
#include <string.h>
void main()
{
    struct student
    {
        int num;
        char name[25];
    };
    student stu;
    stu.num = 123;
    strcpy(stu.name, "John");
    printf("%d - %s", stu.num, stu.name);
}
```

- A. 123 - John
- B. John - John
- C. Compile time error
- D. Runtime error

3.11. Hãy cho biết kết quả của đoạn code sau:

```
#include <stdio.h>
struct Time
{
    int hours;
    int minutes;
    int seconds;
};
int toSeconds(Time now);
void main()
{
    Time t;
    t.hours = 2;
    t.minutes = 30;
    t.seconds = 45;
    printf("Total seconds: %d", toSeconds(t));
}
int toSeconds(Time now)
{
    return 3600 * now.hours + 60 * now.minutes + now.seconds;
}
```

- A. 9045
- B. 3705
- C. 2700
- D. 4800