Bài 1: Viết chương trình khai báo mảng số nguyên có tối đa 20 phần tử.

1. Hàm NhapMang truyền vào tham số số lượng phần tử muốn sử dụng và  
   mảng. Hàm yêu cầu người dùng nhập giá trị phần tử và lưu vào mảng. Hàm  
   không có giá trị trả về. **void NhapMang(int &Soluong, int Arr[])**
2. Hàm XuatMang truyền vào tham số số lượng phần tử đang sử dụng và mảng.  
   Hàm xuất giá trị của tất cả phần tử có trong mảng. Hàm không có giá trị trả  
   về. **void XuatMang(int Soluong, int Arr[])**
3. Liệt kê vị trí và giá trị các phần tử là số chẵn bằng hàm **void lietKePhanTuChan(int Soluong, int Arr[])**
4. Đếm các phần tử có giá trị chia hết cho 5 bằng hàm **void lietKePhanTuChiaHetCho5(int Soluong, int Arr[])**
5. Tính tổng số phần tử nằm trong khoảng min, max bằng hàm **int lietKePhanChiaHetCho5(int Soluong, int Arr[], int min, int max),** yêu cầu nhập min và max
6. Viết hàm **kiemTraSNT(int value)** để kiểm tra xem phải số nguyên tố hay không.
7. Liệt kê các giá trị phần tử là số nguyên tố bằng hàm **void lietKePhanTuLaSNT(int Soluong, int Arr[]).** Sử dụng hàm **kiemTraSNT(int value)** để kiểm tra giá trị từng phần tử.
8. Hàm HoanVi truyền vào 2 tham số số nguyên. Hàm thực hiện đảo giá trị của  
   2 tham số. Hàm không có giá trị. **void HoanVi(int &a, int &b)**
9. Hàm BubbleSort truyền vào tham số mảng. Hàm áp dụng thuật giải Bubble  
   Sort để sắp xếp giá trị các phần tử theo trật tự tăng dần. Hàm này có sử  
   dụng hàm HoanVi khi cần thiết. Hàm không có giá trị trả về.  
   **void BubbleSort (int Soluong, int Arr[])**

* Tại chương trình chính:  
  • Khai báo mảng, biến số lượng phần tử sử dụng.  
  • Yêu cầu người dùng nhập số lượng phần tử muốn sử dụng. Cần đảm bảo số  
  lượng người dùng nhập vào phải ít hơn hoặc bằng số lượng phần tử tối đa  
  được khai báo trong mảng.  
  • Gọi các hàm ở yêu cầu 3, 4, 5, 7. Và gọi lại hàm **XuatMang** để thực hiện kiểm thử chương trình.  
  • Gọi hàm **BubbleSort** truyền vào mảng đã khai báo để sắp xếp trật tự giảm  
  dần.  
  • Gọi hàm XuatMang truyền vào mảng đã khai báo và số lượng phần tử sử  
  dụng để xem kết quả sau khi sắp xếp.

Bài 2:

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

**2.** 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.

**3.** 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.  
o Kiểm tra tính hợp lệ của loại phòng (chỉ chấp nhận “LT” hoặc “TH”), nếu  
không hợp lệ thì yêu cầu nhập lại.  
o Kiểm tra tính hợp lệ của số lượng chỗ ngồi (chỉ chấp nhận số >= 0), nếu  
không hợp lệ thì yêu cầu nhập lại.

**4.** Xây dựng các hàm

• 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 **HienThiPhongHọc** để hiển thị nội dung  
từng phòng học.  
---------------------------------------------------------------------------------------------  
• Hàm **void DSPhongTheoSL(int soluong):** Tìm phòng theo tiêu chí số lượng ≥  
tham số truyền vào. Nếu thoả điều kiện thì gọi hàm HienThiPhongHọc để  
xuất thông tin phòng học đó. Hãy đặt tiêu đề *“DANH SÁCH PHÒNG HỌC CÓ  
SỐ LƯỢNG >= XX CHỖ”* trước khi hiển thị danh sách.

• Hàm **void DSPhongTheoTT(int tt):** Tìm phòng theo tiêu chí tình trạng Bận  
hoặc Rãnh. Nếu thoả điều kiện thì gọi hàm HienThiPhongHọc để xuất thông  
tin phòng học đó. Hãy đặt tiêu đề *“DANH SÁCH PHÒNG HỌC CÓ TÌNH  
TRẠNG YY”* trước khi hiển thi 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 HienThiPhongHọc  
để xuất thông tin phòng học đó. Hãy đặt tiêu đề *“DANH SÁCH PHÒNG HỌC  
ZZ”* trước khi hiển thi 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.  
o Yêu cầu người dùng nhập vào mã phòng học.  
o 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.  
o 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 Rãnh sang Bận, đồng thời xuất thông tin *“Bạn đã  
đặt phòng thành công.”*

* 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:  
  o 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.  
  o 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,gGọi hàm DSPhongTheoSL.  
  o 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.

Bài 3: Xây dựng chương trình tính phương trình bậc hai có dạng ax2 + bx + c = 0;

Yêu cầu:

1. Tạo file **input.txt**

*<Dòng 1: Giá trị của a>  
<Dòng 2: Giá trị của b>*

*<Dòng 3: Giá trị của c>*

1. Dùng con trỏ FILE, đọc file input.txt, viết hàm tìm nghiệm cho phương trình
2. Tạo file **output.txt,** viết kết quả nghiệm ra file trong trường hợp có nghiệm, còn phương trình vô nghiệm ghi ra file kết quả “Vo nghiem”

*<Dòng 1: Giá trị nghiệm 1, hoặc nghiệm kép >  
<Dòng 2: Giá trị nghiệm 2>*