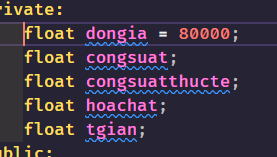
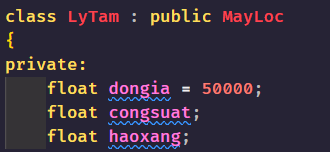
Câu 1:

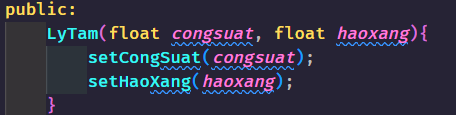
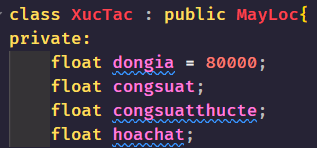
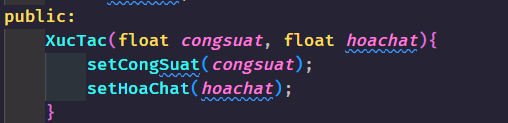
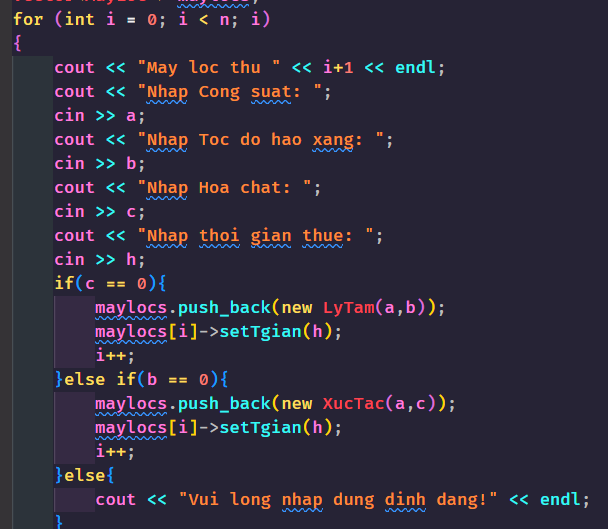
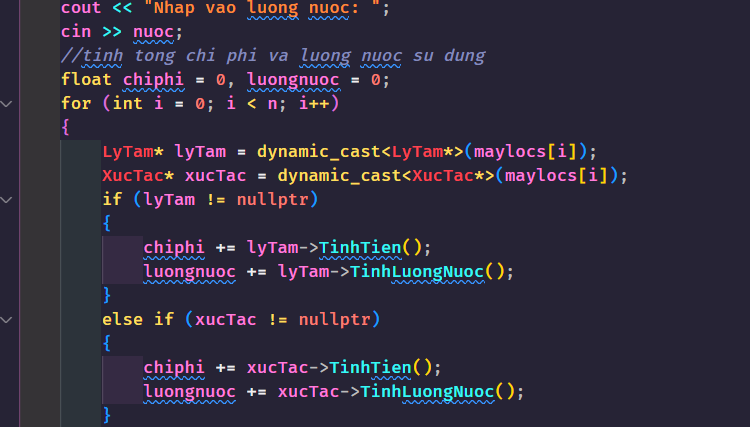
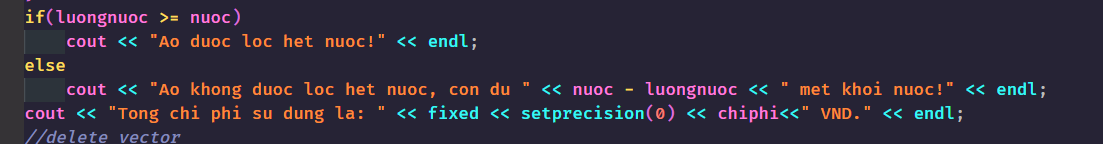
1. Trong hàm main() có 2 constructor ở dòng 02 và dòng 05, 1 destructor dòng 10
2. Lỗi đầu tiên là ở dòng 03, khi pm gọi hàm congSuat(), có khả năng nếu pm vừa khởi tạo, hàm congSuat luôn return về 0, gây ra leak memory; ngoài ra hàm congSuat() nằm trong protected, có nghĩa là nó chỉ được truy cập từ bên trong lớp MayLoc, nếu gọi từ bên ngoài sẽ gay ra lỗi. Lỗi thứ 2 nằm ở dòng 06 và 10, khi pm trỏ về &ml, mà ml không phải một đối tượng được cấp phát bằng new, khả năng delete pm ở dòng 10 sẽ sinh ra lỗi.
3. m\_tgian thuộc private, và lớp MayLoc có sẵn hàm setTgian, vì vậy ta gọi hàm setTgian thay vì gán trực tiếp m\_tgian = t.
4. Sửa hơi nhiều chỗ chứ sửa mỗi /\*2\*/ chạy = niềm tin. Kết quả in ra là : “Luong nuoc = 819”.
5. Dời delete pm lên bên trên dòng pm = &ml;

Câu 2: file 2011C2.cpp  
Đề cho công thức và những cái mà mình nhập vô ghi vào hàm nhập   
vấn đề đầu tiên là máy xúc tác và đơn giá của nó   
muốn gọi kiểu này phải khai báo   
static const float DON\_GIA\_HOA\_CHAT;

Là tên biến   
ở bên ngoài const float MayXucTac::DON\_GIA\_HOA\_CHAT = 10000

Gán giá trị cho nó mới gọi được  
  
muốn lấy giá trị trong private: thì phải có hàm trung gian lấy hàm --setCongSuat( float cs) nó đọc được paravector này float cs gán giá trị công suất của cái class MayXucTac = giá trị của cs là giá trị nhập vào ở đây là temp  
-hàm getCongSuat() float nó sẽ trả về congsuat 2 cái float phải trùng nhau, nó lấy ra get rồi nó trả về tk nào gọi cho nó   
-khi mình cin >> temp nó ghi đè lên temp đỡ phải gọi nhiều biến   
- tính tiền thằng trong hàm main nó không nằm ở class nên cần thằng trung gian để lấy ra, còn ở trong class TinhTien() nólấy thẳng ra được là lấy đơn giá \* cho thời gian mình nhập vào + cho số lượng hóa chất \* với đơn giá hóa chất   
tính công thức thực tế :   
Nếu mà thời gian nhận vào bé hơn 10  
congsuatthucte = congsuat \* (hoachat/ 100)  
  
Khi mà nó chạy hàm này float tinhLuongNuoc() thay đổi giá trị ban đầu công suất thực tế không có gì hết, không có dòng này công suất thực tế bằng 0, dẫn thới đoạn dưới this -> caLCSTT(); sai  
- Khi có công suất thực tế rồi nó sẽ tính công suất lượng nước

Câu 3: file 2011C3.cpp  
Khai báo thư viện dung em dùng <vector> để quản lý nhiều máy bơm dễ hơn, có thể dùng mảng nhưng mảng sẽ rắc rối hơn   
- còn thư viện <iomanip> để fixed lại kết quả đẹp hơn   
  
Hàm Class Mayloc để có them yêu cầu đóng gói, kế thừa.  
- trong hàm class MayLoc có 1 biến thời gian và hai hàm   
 +setTgian(float t) : gián giá trị của m\_tgian = t .  
 +getTgian() : để lấy giá trị nó ra.  
MayLoc() | this->m\_tgian = 0 là hàm bắt đầu khi mà khai báo MayLoc bắt đầu thời gian nó sẽ bằng 0  
  
Hàm virtual ~MayLoc() là 1 hàm hủy ảo để lát nữa mình gọi dynamic\_cast   
  
Class LyTam : Public MayLoc : class LyTam kế thường Public của MayLoc  


Trên Đề cho đơn giá, biến công suất sẽ nhập vào, biến haoxang MayLoc này có tốc độ hao xăng  
  
Hàm này tạo ra MayBom bằng cách bỏ cho nó congsuat và haoxang , thứ mà nó cần thiết  
Hàm TinhTien() và TinhLuongNuoc() theo công thức   
  
  
class XucTac cũng kế thường từ public MayLoc   
và khai báo biến 4 biến, có them congsuatthucte phải tính.  
  
cái xúc tác này tương tự giống với ở LyTam trên   
Những hàm còn lại tương tự các bài trên  
a) hỏi tạo 1 danh Sách N máy lọc  
Tạo 1 biến n ;  
float a,b,c,nuoc;   
a: tượng trưng công suất   
b: tốc độ hao xăng  
c: hóa chất   
nuoc = nước  
float h = 0 là thời gian thuê của từng máy  
Đầu tiên nhập số lượng .  
  
dòng này để khai náo tên vector MayLoc, thuộc về con trỏ máy lọc  
  
  
  
đưa vô vòng lập và nhập từng cái như công suất, hao xang, hóa chất, thời gian thuê;  
- nếu hóa chất = 0 nó là là máy lọc LyTam push\_back một con trỏ mới lúc nó sẽ tạo ra congsuat và haoxang  
-mayloc[i] là máy lọc thứ i, nó sẽ gán thời gian thuê setTgian(h)   
mà máy lọc thứ i là máy lọc push\_back ở trên, sau khi tạo xong thì tăng i++ lên 1.  
  
Nếu mà nhập cả b vs c đều khác 0 thì chương trình sẽ kêu chúng ta nhập lại, nếu không đúng định dạng i sẽ không tăng sẽ, bắt mình nhập lại   
  
  
gọi 2 biến là chiphi và luongnuoc.  
for này phải có i++  
tạo 1 cái con trỏ LyTam nếu mà maylocs[i] thuộc về biến lytam thì lytam sẽ là một con trò khác nullptr  
thì: chiphi tăng 1 cái giá trị của máy lytam ->Tinhtien()  
 tương tự lượng nước cũng vậy  
nếu maylocs[i] không phải là máy xuc tác thí máy lọc xúc tác sẽ trả vè nullptr  
  
  
nếu lượng nước lớn hơn nước