**Bài tập Git**

Sinh viên A:Nguyễn Hữu Danh

Sinh viên B:Nguyễn Chí Cường

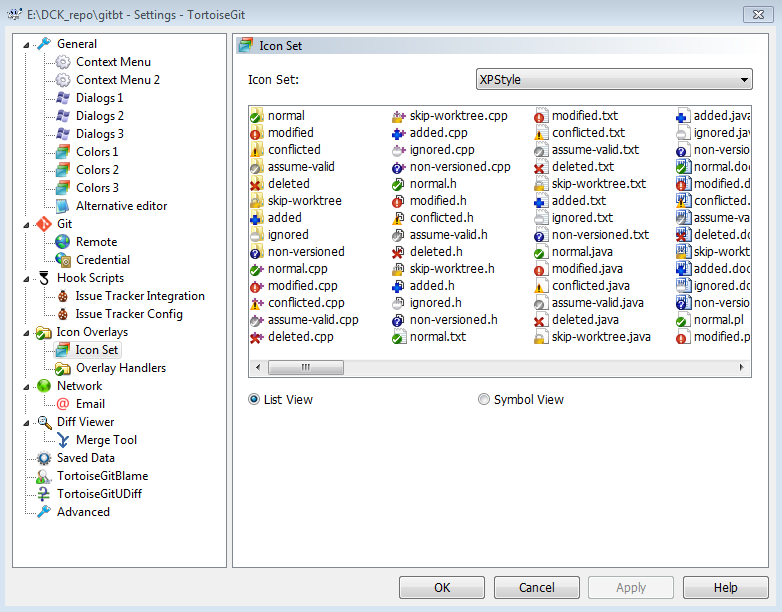
Sinh viên C:Phan Huy Khanh

Sinh viên D:

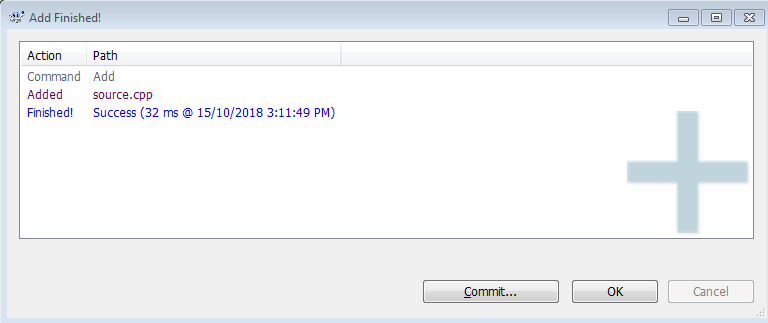
Sinh viên E:

1. Hãy tạo 1 server repository (SR) (sv A)
2. SV A,B,C,D,E tạo 1 thư mục E:\TenNhom\_repo làm local repository (LR)
3. SV A chép file được cung cấp vào LR của mình vừa tạo
4. Nhóm cho biết trạng thái (status) của các file vừa chép vào? Xem trại thái bằng cách nào?

Trả lời:non-versioned

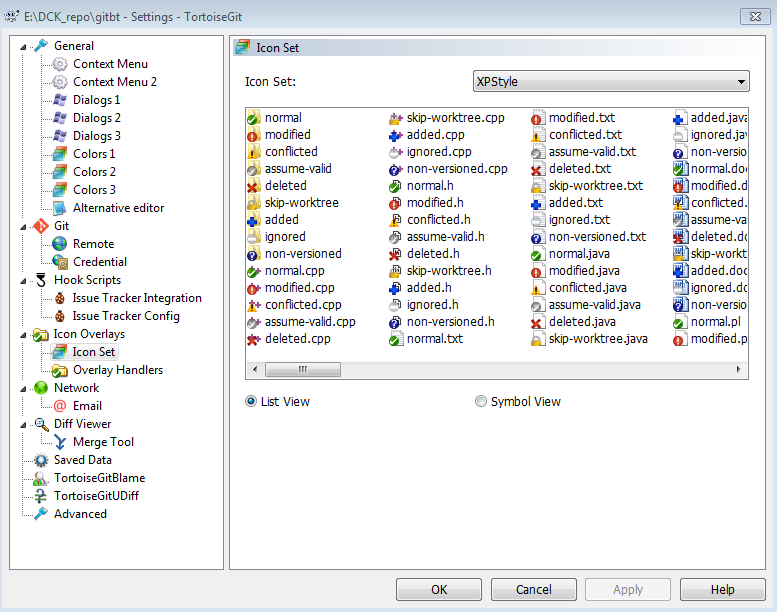


1. SV A hãy **thêm** file đó vào LR

Trả lời:   


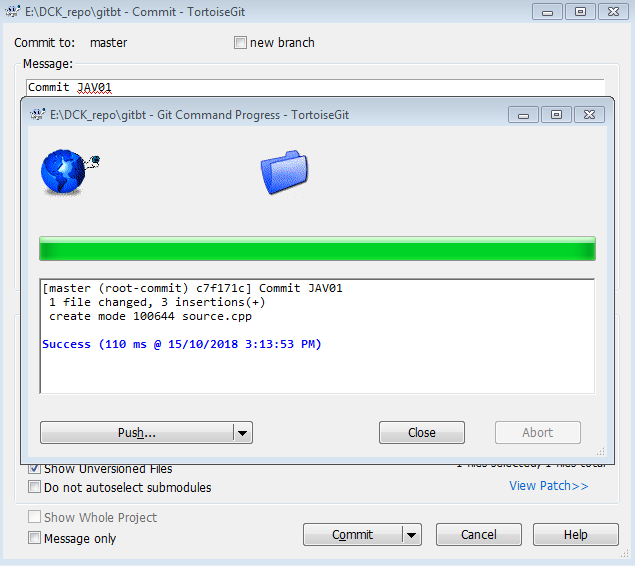
1. Kiểm tra lại trạng thái các file đó

added

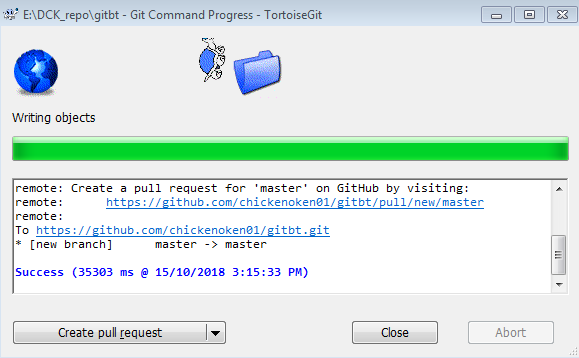


1. **Commit** những thay đổi đó

Trả lời:

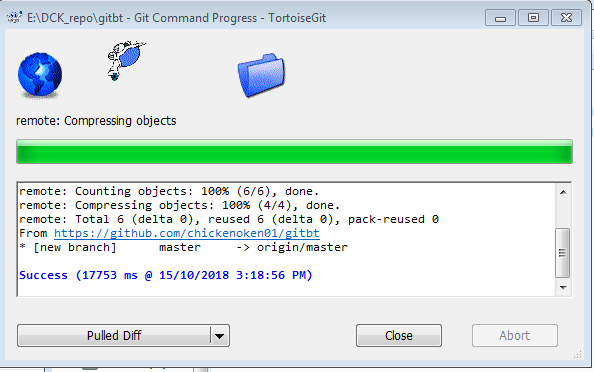


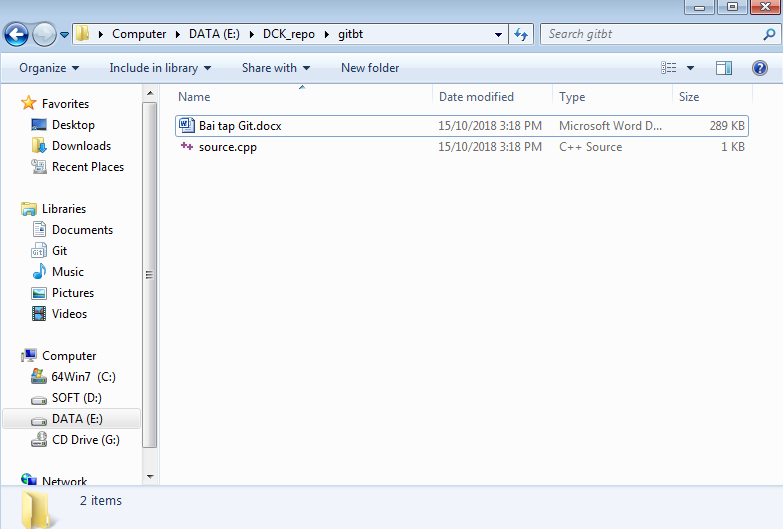
1. Đẩy các thay đổi đó lên SR của nhóm



1. SV B,C thực hiện Pull, SV D,E thực hiện Fetch. Nêu nhận xét

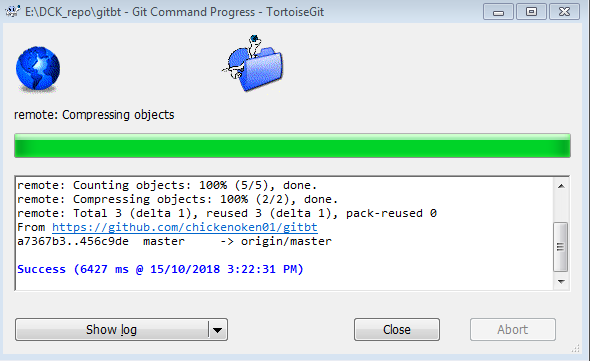
Sv B,C:





Nhận xét : lấy được file mà sinh viên A đã push lên kho .

Sinh viên D,E không có thay đổi



1. Các SV hoàn thiện hàm được cho theo cách giải riêng của mình
2. SV B cập nhật thay đổi lên kho trung tâm SR

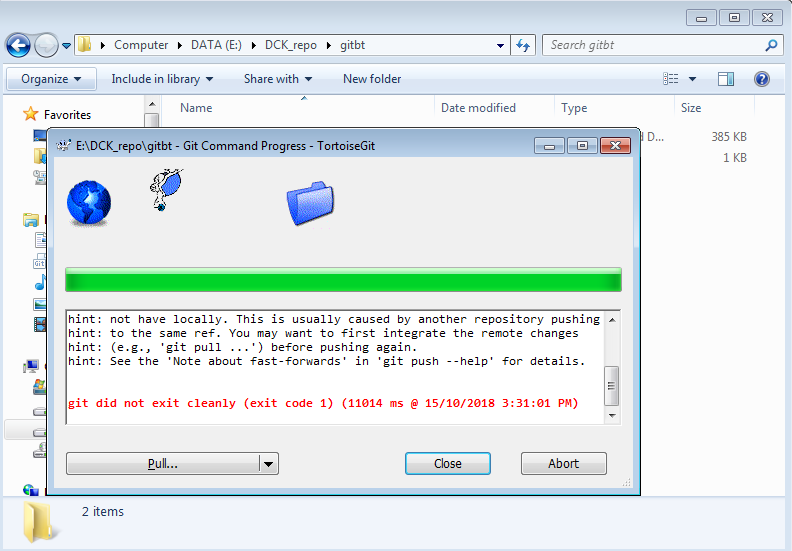
B Push lên kho bình thường không xảy ra lỗi

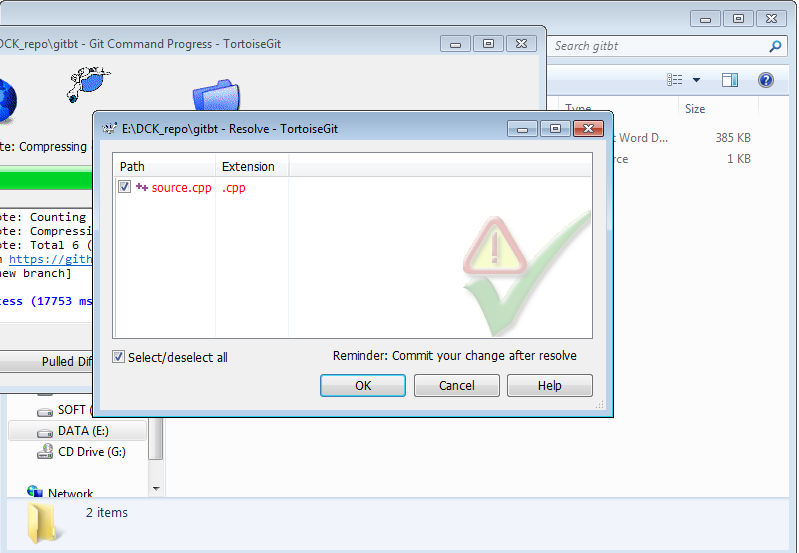
1. SV C cập nhật thay đổi lên kho trung tâm SR

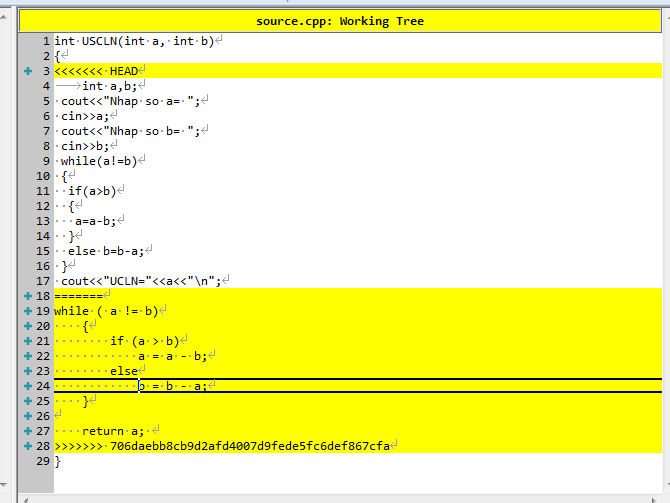
C không push lên được kho , xảy ra lỗi xung đột .

1. Nhóm nêu cách giải quyết xung đột

Đầu tiên : pull phiên bản mới nhất để đồng bộ dữ liệu , xuất hiện ra nhiều file , sau đó resolve để lựa chọn thay đổi khác nhau của file , commit và push lên lại kho







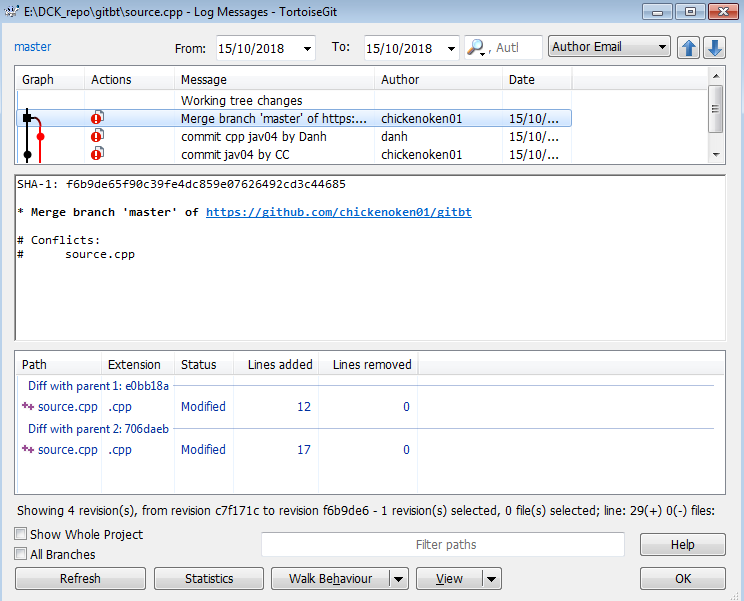
1. SV D,E cập nhật lại LR của mình để có được phần mới bổ sung của SV B và C
2. SV D or A cho biết những chỗ khác nhau của file trong phiên bản hiện tại với phiên bản trước.

Có sự thay đổi so với bản trước đó.



1. SV D or A cho biết source.cpp được cập nhật bởi ai và vào lúc nào?

Được thay đổi bởi chickenoken01 ( CC ) vào thời giao … trong hình.



1. SV A,B,C,D,E thêm nhánh mới có tên boi\_so\_chung. Viết hàm tìm BSC lớn nhất
2. SV A,B,C, D, E trộn nhánh boi\_so\_chung vào master
3. Quay về trước phiên bản trước khi có nhánh boi\_so\_chung