SVN - FIX MISTAKES

Suppose *Jerry* accidently modifies *array.c* file and he is getting compilation errors. Now he wants to throw away the changes. In this situation, 'revert' operation will help. Revert operation will undo any local changes to a file or directory and resolve any conflicted states.

```
[jerry@CentOS trunk]$ svn status
```

Above command will produce the following result.

```
M       array.c
```

Let's try to make array as follows:

```
[jerry@CentOS trunk]$ make array
```

Above command will produce the following result.

```
cc     array.c   -o array
array.c: In function ‘main’:
array.c:26: error: ‘n’ undeclared (first use in this function)
array.c:26: error: (Each undeclared identifier is reported only once
array.c:26: error: for each function it appears in.)
array.c:34: error: ‘arr’ undeclared (first use in this function)
make: *** [array] Error 1
```

Jerry performs 'revert' operation on *array.c* file.

```
[jerry@CentOS trunk]$ svn revert array.c
Reverted 'array.c'

[jerry@CentOS trunk]$ svn status
[jerry@CentOS trunk]$
```

Now compile the code.

```
[jerry@CentOS trunk]$ make array
cc     array.c   -o array
```

After the revert operation, his working copy is back to its original state. Revert operation can revert a single file as well as a complete directory. To revert a directory, use -R option as shown below.

```
[jerry@CentOS project_repo]$ pwd
/home/jerry/project_repo

[jerry@CentOS project_repo]$ svn revert -R trunk
```

Till now, we have seen how to revert changes, which has been made to the working copy. But what if you want to revert a committed revision! Version Control System tool doesn't allow to delete history from the repository. We can only append history. It will happen even if you delete files from the repository. To undo an old revision, we have to reverse whatever changes were made in the old revision and then commit a new revision. This is called a reverse merge.

Let us suppose Jerry adds a code for linear search operation. After verification he commits his changes.

```
[jerry@CentOS trunk]$ svn diff
Index: array.c
===================================================================
```

```
--- array.c   (revision 21)
+++ array.c   (working copy)
@@ -2,6 +2,16 @@

 #define MAX 16

+int linear_search(int *arr, int n, int key)
+{
+   int i;
+
+   for (i = 0; i < n; ++i)
+      if (arr[i] == key)
+         return i;
+   return -1;
+}
+
 void bubble_sort(int *arr, int n)
 {
    int i, j, temp, flag = 1;

[jerry@CentOS trunk]$ svn status
?       array
M       array.c

[jerry@CentOS trunk]$ svn commit -m "Added code for linear search"
Sending        trunk/array.c
Transmitting file data .
Committed revision 22.
```

Jerry is curious about what Tom is doing. So he checks the Subversion log messages.

```
[jerry@CentOS trunk]$ svn log
```

The above command will produce the following result.

```
------------------------------------------------------------------------
r5 | tom   | 2013-08-24 17:15:28 +0530 (Sat, 24 Aug 2013) | 1 line

Add binary search operation
------------------------------------------------------------------------
r4 | jerry | 2013-08-18 20:43:25 +0530 (Sun, 18 Aug 2013) | 1 line

Add function to accept input and to display array contents
```

After viewing the log messages, Jerry realizes that he did a serious mistake. Because Tom already implemented binary search operation, which is better than the linear search; his code is redundant, and now Jerry has to revert his changes to the previous revision. So, first find the current revision of the repository. Currently, the repository is at revision 22 and we have to revert it to the previous revision, i.e. revision 21.

```
[jerry@CentOS trunk]$ svn up
At revision 22.

[jerry@CentOS trunk]$ svn merge -r 22:21 array.c
--- Reverse-merging r22 into 'array.c':
U    array.c

[jerry@CentOS trunk]$ svn commit -m "Reverted to revision 21"
Sending        trunk/array.c
Transmitting file data .
Committed revision 23.
```