1. copy everything in LabQ into Dir1 with cp. Keep owner and date info. term LabQ will not be copied but everything inside it (recursively). Use relative addressing
   cp -Rp . ../Answers/Dir1

2. copy everything in the current directory into Dir2 using tar without using any explicit tar file, if possible. You can use and explicit tar file, but you will lose credit.
   tar cf - . | (cd ../Answers/Dir2; tar xf - )
   tar cf - . | tar xf -C ../Answers/Dir2
   tar cf ../Answers/LabQ.tar; tar xf ../Answers/LabQ.tar -C ../Answers/Dir2

3. copy everything in LabQ into Dir3 with cpio with using an auxiliary cpio file
   find . -depth | cpio -o > LabQ.cpio; (cd ../Answers/Dir3; cpio -i < ../LabQ.cpio)

4. find all files related with program tar
   man -k tar > ../Answers/4.txt

5. Find list of all *.txt files which are newer than time.stamp
   find . -type f -newer time.stamp -name "*.txt" > ../Answers/5.txt

6. combine all *.txt files within LabQ hierarchy
   find . -type f -name "*.txt" | xargs cat > ../Answers/6.txt

7. find all *.txt files whose names start with a-k within LabQ hierarchy and copy them into Dir4 while retaining directory structure
   find . -type f -depth -name "[a-k]*.txt" | cpio -pd ../Answers/Dir4

8. In a single command create directories A B C D each containing E F H, and each of which containing 1 2 3 4 5, each of which containing X Y and Z in directory Answers (we need to see directory and corresponding command). Also in a single command create empty files x in X's, y in Y's and z in Z's.
   mkdir -p ../Answers/A,B,C,D/E,F/H/1,2,3,4,5/X,Y,Z
touch ../Answers/A,B,C,D/E,F/H/1,2,3,4,5/X/x,Y/y,Z/z

9. Consider all files in the current directory: including sub directories
   - determine all lines among all files which contains string ayse or elif, case insensitive
   egrep -irh "elif|ayse" . > ../Answers/9x.txt
   restricting search to all *.txt files:
   find . -type f -name "*.txt" | xargs cat | egrep -i "elif|ayse" > ../Answers/9.txt
   find . -type f -name "*.txt" -exec egrep -i "elif|ayse" ' ' \; > ../Answers/9.txt

10. Consider all *.txt files in LabQ (on the surface) determine files which contains elif followed by ayse case insensitive on a line, (under 10.txt) and combine them (under 10.TXT)
    grep -il "elif.*ayse" *.txt > ../Answers/10.txt
grep -il "elif.*ayse" *.txt | xargs cat > ../Answers/10.TXT

06/April/2015  Solution CTIS166 Quiz 3  1
11. find all *.txt files within LabQ hierarchy and put these files in TXT1.zip
   find . -type f -name "*.txt" | xargs zip ../Answers/TXT1

12. Extract contents of AB.tar.gz into Dir6 without using gunzip or zcat;i.e just tar with suitable options only You issue commands within LabQ.
   tar xzf AB.tar.gz -C ../Answers/Dir6
   (cd ../Answers/Dir6; tar xzf ../LabQ/AB.tar.gz )

13. Find lines in Dene3.txt.bz2 containing string elif and ayse case insensitive without using bunzip2 and bzcat
   -i elif Dene3.txt.bz2 | grep -i ayse > ../Answers/13.txt

14. Copy ABC and xetc directories into Dir8, and compress all ordinary files in Dir8 with xz
   cp -Rp ABC xetc ../Answers/Dir8; find ../Answers/Dir8 -type f -print -exec xz '' 

15. Add execute permissions for file exec.SH to all , remove write from others, add to write and suid property to group; all in a single statement, and use symbolic method (Do these after copying into Answers)
   cp exec.SH ../Answers; chmod a+x,o-w,g+ws ..../Answers/exec.SH

16. Consider Dene.txt, remove first 25 lines and last 15 lines obtaining DENE.TXT. Do not use information about size of Dene.txt, just assume it has more than 60 lines in it. We need to see original line numbers. Just use filters. you may use tac
   cat -n Dene.txt | tail -n +26 | head -n -15 > ../Answers/DENE.TXT
   cat -n Dene.txt | tail -n +26 | tac | tail -n +26 | tac > ../Answers/DENE.TXT
   cat -n Dene.txt | head -n -15 | tac | head -n +25 | tac > ../Answers/DENE.TXT

17. Given a.txt, we want to determine the lines containing string Net and put them in files 17-1.txt, 17-2.txt and 17-3.txt. How would you do it using:(each

   grep: grep Net a.txt > ../Answers/17-1.txt
   sed: sed -n '/Net/p' a.txt > ../Answers/17-2.txt
   awk: awk '/Net/' a.txt > ../Answers/17-3.txt

18. Given a.txt, we want to determine non-empty lines and put them in a files 18-1.txt, 18-2.txt, and 18-3.txt. How would you do it using: (each

   grep: grep -v ^$ a.txt > ../Answers/18-1.txt
   sed: sed '/^$/d' a.txt > ../Answers/18-2.txt
   sed -n '/^$/!p' a.txt > ../Answers/18-2.txt
   awk: awk '/^$/!' a.txt > ../Answers/18-2.txt

19. Given a.txt, we want to determine the lines containing string Net and not containing Fox and put them in files 19-1.txt, 19-2.txt and 19-3.txt How would you do it using:(each

   grep: grep Net a.txt | grep -v Fox > ../Answers/19-1.txt
   sed: sed '/Net/p' a.txt | sed '/Fox/d' > ../Answers/19-2.txt
   awk: awk '/Net/' a.txt | awk '/Fox/' > ../Answers/19-3.txt