**CMD Command**

* Your first challenge is to print "hello world" on the terminal in a single command.
* echo "hello world"
* echo hello world
* echo 'hello world'
* printf "hello world"
* echo hello\ world
* printf 'hello world'
* echo "hello world";
* 'echo' hello world
* 'echo' "hello world"
* echo " hello world"
* echo -n "hello world"
* echo "hello world "
* printf "hello world\n"
* echo "hello world"
* echo 'hello world';
* cat <<< "hello world"
* echo ""hello world""
* printf "%s" "hello world"
* echo ''hello world''
* echo " hello world "
* echo "hello world" > hello.txt && cat hello.txt
* echo hello world
* echo -e "hello world"
* printf "hello world";
* echo -n hello world
* echo hello world;
* echo 'hello world'
* echo 'hello world '
* echo $"hello world"
* printf '%s' "hello world"
* echo "uryyb jbeyq" | tr '[n-za-m]' '[a-z]'
* echo "hello world" >> texto.txt | cat texto.txt
* echo ""hello world
* "echo" hello world
* printf -- "hello world\n"
* echo ' hello world'
* 'echo' 'hello world'
* echo -e hello world
* echo "hello world"\
* printf "%s %s" hello world
* printf '%s\n' 'hello world'
* 'echo' "hello world";
* echo -ne "hello world"
* echo -e ''hello world''
* echo "hello world" >> file1 && cat file1
* echo 'hello world' ;
* command echo hello world
* printf '%s %s\n' hello world
* echo "hello world"; touch text.txt
* echo "hello "world
* Print the current working directory.
* pwd
* echo $PWD
* dirs
* pwd dir
* pwd current working directory
* echo $(pwd)
* pwd ls
* pwd hello world
* pwd "working directory"
* pwd "hello world"
* pwd working directory
* pwd echo
* pwd print
* echo "$PWD"
* pwd "current working directory"
* pwd command
* printf $PWD
* pwd .
* echo | pwd
* pwd directory
* pwd c
* pwd [-LP]
* pwd\
* echo `pwd`
* pwd home
* dirs -p
* pwd /home
* pwd "home"
* printf "$PWD"
* pwd $
* pwd root
* pwd dirs
* "pwd"
* pwd working
* pwd C
* pwd p
* pwd -L
* printf "${PWD}"
* pwd /var/challenges/create\_directory
* pwd \
* pwd 1s
* echo &pwd
* realpath ./
* echo $(pwd);
* pwd C\Program Files
* pwd P
* pwd -P
* pwd command\_line\_for\_the\_win
* 'pwd'
* pwd -LP
* List names of all the files in the current directory, one file per line.
* ls
* ls -1
* ls -A
* ls \*
* ls /var/challenges/list\_files
* ls -h
* ls .
* ls -L
* dir -1
* ls $pwd
* ls -r
* ls | cat
* ls -c
* ls -p
* ls -p | grep -v /
* ls ./
* ls $PWD
* ls -F
* ls -t
* for f in \*;do echo "$f";done
* find \*
* ls -v
* ls -l | awk '{print $9}'
* ls | more
* ls \
* for FILE in \*; do echo $FILE; done
* ls\
* ls -q
* ls -S
* ls|cat
* ls | sort
* ls -H
* ls -A1
* ls -b
* printf '%s\n' \*
* ls -1 .
* for f in \*;do echo $f;done
* ls -N
* ls |cat
* for f in \*; do echo $f; done
* for f in \*; do echo "$f";done
* find . -type f| cut -d "/" -f2
* ls -1A
* ls ; echo ""
* ls | tr " " "\n"
* for i in \*; do echo $i; done
* ls -C1
* ls -ch
* ls -AA
* ls | tr ' ' '\n'
* There is a file named access.log in the current directory. Print the contents.
* cat access.log
* cat \*
* cat "access.log"
* cat ./access.log
* tail access.log
* head access.log
* echo "`<access.log`"
* cat access.log
* pg access.log
* cat 'access.log'
* cat < access.log
* cat a\*
* cat access\*
* awk '{print $0}' access.log
* echo "$(<access.log)"
* cat acc\*
* while read line; do echo $line; done < access.log
* cat /var/challenges/print\_file\_contents/access.log
* cat \*log
* cat \*.log
* cat access.log\
* cat ac\*
* cat acces\*
* cat $(pwd)/access.log
* echo "$(< access.log)"
* grep "" access.log
* echo "$(<access.log )"
* cat -t access.log
* cat -t "access.log"
* cat a\*g
* cat acce\*
* cat access.log;
* tail access.log
* cat \*access.log\*
* grep . access.log
* cat ''access.log''
* cat access.log\*
* cat a\*log
* cat <access.log
* cat -v access.log
* cat -- access.log
* cat access.l\*
* 'cat' access.log
* cat access.log |tail
* grep '\_\*' access.log
* grep ".\*" access.log
* cat access.\*
* tail ./access.log
* cat access.log
* grep '.\*' access.log
* Print the last 5 lines of "access.log".
* tail -5 access.log
* tail -n 5 access.log
* cat access.log | tail -5
* tail -n5 access.log
* cat access.log | tail -n 5
* tail access.log -n 5
* tail -5 "access.log"
* tail -5 access.log
* tail -5 ./access.log
* cat access.log |tail -5
* cat access.log | tail -n5
* cat access.log|tail -5
* tail --lines=5 access.log
* tail -n 5 "access.log"
* tail -n 5 access.log
* tail -n 5 ./access.log
* tail -n -5 access.log
* tail -5 \*
* tail access.log -n5
* tail --lines 5 access.log
* tail -5 'access.log'
* cat access.log| tail -5
* tail -5 access.log
* sed -n '6,10p' access.log
* cat access.log|tail -n 5
* sed -n 6,10p access.log
* awk '{if(NR>5) print $0}' access.log
* cat "access.log" | tail -5
* tail -5 a\*
* tail -n5 \*
* tail -5 acc\*
* cat access.log |tail -n 5
* cat access.log | tail -n -5
* cat ./access.log | tail -5
* 'tail' -5 access.log
* tail access.log --lines=5
* tail -5 \*g
* tail -5 \*.log
* tail -5 access\*
* tail -n5 access.log
* tail access.log -n -5
* cat access.log | tail -5f
* cat access.log | tail -5
* tail -n 5 /var/challenges/last\_lines/access.log
* tail -n 5 ac\*
* tail -n 5 \*.log
* tail -5l access.log
* tail -n 5 access.log
* tail --line=5 access.log
* cat access.log| tail -n 5
* Create an empty file named take-the-command-challenge in the current working directory.
* touch take-the-command-challenge
* cat > take-the-command-challenge
* touch "take-the-command-challenge"
* cd > take-the-command-challenge
* touch ./take-the-command-challenge
* touch take-the-command-challenge
* >take-the-command-challenge
* touch 'take-the-command-challenge'
* cat >take-the-command-challenge
* echo -n > take-the-command-challenge
* cd> take-the-command-challenge
* cat>take-the-command-challenge
* cat >> take-the-command-challenge
* cat > "take-the-command-challenge"
* touch take-the-command-challenge .
* cat> take-the-command-challenge
* type nul > take-the-command-challenge
* nul > take-the-command-challenge
* echo. > take-the-command-challenge
* file > take-the-command-challenge
* touch named take-the-command-challenge
* touch > take-the-command-challenge
* cd >take-the-command-challenge
* cd. > take-the-command-challenge
* touch take-the-command-challenge ./
* cat "" > take-the-command-challenge
* >> take-the-command-challenge
* cd>take-the-command-challenge
* echo -n >take-the-command-challenge
* touch "take-the-command-challenge" .
* echo.>take-the-command-challenge
* cat > take-the-command-challenge
* cd. >take-the-command-challenge
* touch take-the-command-challenge;
* touch take-the-command-challenge\
* tee take-the-command-challenge
* "" > take-the-command-challenge
* : > take-the-command-challenge
* cat > take-the-command-challenge
* null > take-the-command-challenge
* touch> take-the-command-challenge
* :>take-the-command-challenge
* type NUL > take-the-command-challenge
* "">take-the-command-challenge
* :> take-the-command-challenge
* >./take-the-command-challenge
* nul>take-the-command-challenge
* >take-the-command-challenge
* touch take-the-command-challenge
* >>take-the-command-challenge
* Create a directory named tmp/files in the current working directory
* mkdir -p tmp/files
* mkdir tmp tmp/files
* mkdir tmp && mkdir tmp/files
* mkdir tmp/ tmp/files
* mkdir tmp && cd tmp && mkdir files
* mkdir tmp ; cd tmp ; mkdir files
* mkdir -p ./tmp/files
* mkdir tmp; mkdir tmp/files
* mkdir tmp/files -p
* mkdir tmp; cd tmp; mkdir files
* mkdir "tmp/" "tmp/files"
* mkdir tmp/ && mkdir tmp/files
* mkdir -p tmp/files
* mkdir tmp mkdir tmp/files
* mkdir tmp touch tmp/files
* mkdir -p tmp/files/
* mkdir tmp ; mkdir tmp/files
* mkdir "tmp" "tmp/files"
* mkdir -p "tmp/files"
* mkdir tmp;mkdir tmp/files
* mkdir tmp tmp/files
* mkdir tmp/ mkdir tmp/files
* mkdir -p tmp tmp/files
* mkdir tmp/ & mkdir tmp/files
* mkdir tmp/ touch tmp/files
* mkdir tmp
* cd tmp
* mkdir files
* mkdir -pv tmp/files
* mkdir tmp;cd tmp;mkdir files
* mkdir ./tmp && mkdir ./tmp/files
* mkdir tmp && mkdir ./tmp/files
* mkdir tmp ./tmp/files
* mkdir {tmp,tmp/files}
* mkdir --parents tmp/files
* mkdir tmp/; mkdir tmp/files
* mkdir 'tmp' 'tmp/files'
* mkdir 'tmp/' 'tmp/files'
* mkdir ./tmp ./tmp/files
* mkdir tmp|mkdir tmp/files
* mkdir -p tmp/files
* mkdir tmp/ ; mkdir tmp/files
* mkdir tmp && cd tmp/ && mkdir files
* mkdir tmp tmp/files/
* mkdir {tmp/,tmp/files}
* mkdir tmp/ && cd tmp && mkdir files
* mkdir tmp tmp//files
* mkdir tmp/ && cd tmp/ && mkdir files
* mkdir tmp/ tmp/files/
* mkdir -p ./tmp/files/
* mkdir tmp
* mkdir tmp/files
* mkdir -vp tmp/files
* Copy the file named take-the-command-challenge to the directory tmp/files
* cp take-the-command-challenge tmp/files
* cp take-the-command-challenge tmp/files/
* cp take-the-command-challenge tmp/files/take-the-command-challenge
* cp take-the-command-challenge ./tmp/files
* cp "take-the-command-challenge" tmp/files
* cp take-the-command-challenge tmp/files
* cp take-the-command-challenge tmp/files/.
* cp take-the-command-challenge ./tmp/files/
* cp -r take-the-command-challenge tmp/files
* cp take\* tmp/files
* cp ./take-the-command-challenge ./tmp/files
* cp take-the-command-challenge tmp/files
* cp take-the-command-challenge mv tmp/files
* cp ./take-the-command-challenge tmp/files
* cp "take-the-command-challenge" "tmp/files"
* cp 'take-the-command-challenge' tmp/files
* touch tmp/files/take-the-command-challenge
* cp take-the\* tmp/files
* cp take-the-command-challenge to tmp/files
* cp take-the-command-challenge ./tmp/files/take-the-command-challenge
* cp take-the-command-challenge -t tmp/files
* cp ./take-the-command-challenge ./tmp/files/
* cp -p take-the-command-challenge tmp/files
* cp take-the-command-challenge cd tmp/files
* cp named take-the-command-challenge tmp/files
* cp tak\* tmp/files
* cp take-the-command-challenge "tmp/files"
* cp take-the-command-challenge / tmp/files
* cp take-the-command-challenge -p tmp/files
* cp 'take-the-command-challenge' tmp/files/
* cp take-the-command-challenge > tmp/files/take-the-command-challenge
* cp ./take-the-command-challenge ./tmp/files/take-the-command-challenge
* cp t\* tmp/files
* cp take-the-command-challenge . tmp/files
* cp 'take-the-command-challenge' ./tmp/files
* cat take-the-command-challenge > tmp/files/take-the-command-challenge
* cp take-\* tmp/files
* cp ./take-the-command-challenge tmp/files/
* cp ta\* tmp/files
* cp \*take\* tmp/files
* cp take-the-command-challenge tmp/files/
* cp take-the-command-challenge -r tmp/files
* cp take-the-command-challenge\* tmp/files
* cp take-the-command-challenge tmp/files/
* cp -a take-the-command-challenge tmp/files
* mkdir -p tmp/files && cp take-the-command-challenge tmp/files
* cp take-the-command-challenge /var/challenges/copy\_file/tmp/files
* cp ./take-the-command-challenge tmp/files/take-the-command-challenge
* cp take-the-command-challenge tmp/files -r
* cp -R take-the-command-challenge tmp/files
* Move the file named take-the-command-challenge to the directory tmp/files
* mv take-the-command-challenge tmp/files
* mv take-the-command-challenge tmp/files/
* mv take-the-command-challenge tmp/files/take-the-command-challenge
* mv take-the-command-challenge ./tmp/files
* mv "take-the-command-challenge" tmp/files
* mv take\* tmp/files
* mv take-the-command-challenge tmp/files
* mv take-the-command-challenge tmp/files/.
* mv take-the-command-challenge ./tmp/files/
* mv ./take-the-command-challenge ./tmp/files
* mv take-the-command-challenge tmp/files
* mv "take-the-command-challenge" "tmp/files"
* mv ./take-the-command-challenge tmp/files
* mv -f take-the-command-challenge tmp/files
* mv 'take-the-command-challenge' tmp/files
* mv take-the-command-challenge -t tmp/files
* mv take-the-command-challenge to tmp/files
* mv take-the\* tmp/files
* mv take-the-command-challenge ./tmp/files/take-the-command-challenge
* mv ./take-the-command-challenge ./tmp/files/
* mv take-the-command-challenge "tmp/files"
* mv named take-the-command-challenge tmp/files
* mv tak\* tmp/files
* mv take-the-command-challenge tmp/files
* mv -f take-the-command-challenge tmp/files/
* mv take-\* tmp/files
* mv take-the-command-challenge cd tmp/files
* mv ta\* tmp/files
* mv take-the-command-challenge . tmp/files
* mv take-the-command-challenge tmp/files/
* mv 'take-the-command-challenge' 'tmp/files'
* mv t\* tmp/files
* mv take\* tmp/files/
* mv take-the-command-challenge / tmp/files
* mv ./take-the-command-challenge ./tmp/files/take-the-command-challenge
* mv 'take-the-command-challenge' tmp/files/
* mv -i take-the-command-challenge tmp/files
* mv take-the-command-challenge dir tmp/files
* mv "take-the-command-challenge" tmp/files/
* mv \*take\* tmp/files
* mv take-the-command-challenge /var/challenges/move\_file/tmp/files
* mv ta\* tmp/files/
* mv take\* ./tmp/files
* mv ./take-the-command-challenge tmp/files/
* mv take-the-command-challenge\* tmp/files
* mv -v take-the-command-challenge tmp/files
* mv take-the-command-challenge tmp/files -f
* mv ./take-the-command-challenge tmp/files/take-the-command-challenge
* mv tak\* ./tmp/files/
* mv take-the-command-challenge -f tmp/files
* A symbolic link is a type of file that is a reference to another file. Create a symbolic link named take-the-command-challenge that points to the file tmp/files/take-the-command-challenge.
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge
* ln -s tmp/files/\*
* ln -s tmp/files/take-the-command-challenge "take-the-command-challenge"
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s ./tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s "tmp/files/take-the-command-challenge" "take-the-command-challenge"
* ln -s tmp/files/take-the-command-challenge .
* ln -sf tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* cp -s tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s ./tmp/files/take-the-command-challenge ./take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge ./take-the-command-challenge
* ln -s tmp/files/take\*
* ln -s tmp/files/take-the-command-challenge
* ln -s "tmp/files/take-the-command-challenge"
* ln --symbolic tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s ./tmp/files/take-the-command-challenge
* ln -s -T tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge ./
* ln tmp/files/take-the-command-challenge take-the-command-challenge -s
* ln -s "tmp/files/take-the-command-challenge" take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge 'take-the-command-challenge'
* ln --symbolic tmp/files/take-the-command-challenge
* ln -sf tmp/files/take-the-command-challenge
* ln tmp/files/take-the-command-challenge -s take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* touch ./tmp/files/take-the-command-challenge | ln -s ./tmp/files/take-the-command-challenge
* ln -s tmp/files/ta\*
* ln -s ./tmp/files/take-the-command-challenge "take-the-command-challenge"
* ln -s /var/challenges/create\_symlink/tmp/files/take-the-command-challenge
* cp -s tmp/files/take-the-command-challenge .
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* ln --symbolic ./tmp/files/take-the-command-challenge ./take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* cp -s take-the-command-challenge tmp/files/take-the-command-challenge .
* ln -s tmp/files/\*
* cp -s tmp/files/tak\* .
* ln -sT tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s tmp/files/take\* .
* ln -s tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s ./tmp/files/take-the-command-challenge take-the-command-challenge
* ln tmp/files/take-the-command-challenge . -s && ls -la
* ln -ns tmp/files/take-the-command-challenge take-the-command-challenge
* ln -fs tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s -f tmp/files/take-the-command-challenge take-the-command-challenge
* ln -s tmp/\*/\* take-the-command-challenge
* ln -s tmp/files/take-the-command-challenge\
* ln -s tmp/files/take-the-command-challenge
* ln -sf tmp/files/take-the-command-challenge
* Delete all of the files in this challenge directory including all subdirectories and their contents.
* rm -rf \* .\*
* find . -delete
* rm -r \* .\*
* find -delete
* rm -rf .\* \*
* find . -mindepth 1 -delete
* rm -rf {\*,.\*}
* rm -r .\* \*
* rm -rf \* && rm -rf .\*
* find . -exec rm -rf {} \;
* rm -rf -- ..?\* .[!.]\* \*
* ls -a | xargs rm -rf
* rm -rf ..?\* .[!.]\* \*
* rm -rf {,.[!.],..?}\*
* rm -fr \* .\*
* rm -rv /var/challenges/delete\_files && mkdir /var/challenges/delete\_files
* rm -rf \* .\*
* find . -delete
* rm -rf $(ls -a)
* ls -A1 | xargs rm -rf
* rm -R \* .\*
* rm -rf ./{\*,.\*}
* find . -type f,d -delete
* rm -rf ./\* ./.\*
* find ./ -delete
* find /var/challenges/delete\_files -mindepth 1 -delete
* find . | xargs rm -rf
* rm -rf /var/challenges/delete\_files/{\*,.\*}
* find . -name "\*" -delete
* rm -rfv {\*,.\*}
* rm -r $(ls -a)
* rm -rfv \* .\*
* rm -rf ./\* .\*
* find . \* -delete
* find -mindepth 1 -delete
* rm -rfv /var/challenges/delete\_files/{\*,.\*}
* rm -rf \* .\*
* find . -name . -o -prune -exec rm -rf -- {} +
* ls -A | xargs rm -rf
* rm -r \* .\*
* rm -r \* && rm -r .\*
* rm -R .\* \*
* rm -r ./\* .\*
* ls -a | xargs rm -r
* rm -Rf \* .\*
* rm -rf ./.\* ./\*
* rm -fr .\* \*
* rm -dr \* .\*
* rm -r \* / .\*
* rm .\* \* -r
* rm -Rf .\* \*
* There are files in this challenge with different file extensions. Remove all files with the .doc extension recursively in the current working directory.
* rm \*\*/\*.doc
* find . -name "\*.doc" -delete
* rm -rf \*\*/\*.doc
* rm -r \*\*/\*.doc
* find . -name "\*.doc" -type f -delete
* find . -type f -name "\*.doc" -delete
* find . -name '\*.doc' -delete
* find -name "\*.doc" -delete
* find . -type f -name "\*.doc" -exec rm {} \;
* rm \*\*/\*doc
* find . -type f -name '\*.doc' -delete
* find -name '\*.doc' -delete
* find . -type f -name "\*.doc" -exec rm {} +
* find . -name "\*.doc" | xargs rm
* rm -r \*\*/\*doc
* rm \*\*/\*\*.doc
* find -type f -name "\*.doc" -delete
* find . -name '\*.doc' -type f -delete
* rm -rf \*\*/\*doc
* find . -name "\*.doc" -exec rm {} \;
* find . -type f -name "\*.doc" -exec rm -f {} \;
* rm \*\*/\*.doc
* find -name "\*.doc" -type f -delete
* find . -iname "\*.doc" -delete
* find . -name '\*.doc' -exec rm {} \;
* rm -f \*\*/\*.doc
* find ./ -name "\*.doc" -delete
* rm -R \*\*/\*.doc
* rm -rf ./\*\*/\*.doc
* find . -name "\*doc" -delete
* find -type f -name '\*.doc' -delete
* rm -rf \*\*/\*\*.doc
* rm -fr \*\*/\*.doc
* find . -name "\*.doc" -exec rm -rf {} \;
* rm -rv \*\*/\*.doc
* find -name "\*.doc" | xargs rm
* rm -r \*\*/\*.doc
* rm \*\*/\*.doc -r
* find . -type f -name "\*.doc" -exec rm -rf {} \;
* find . -name '\*doc' -delete
* find . -iname '\*.doc' -delete
* rm \*\*/\*doc -r
* rm -v \*\*/\*.doc
* find . -name \\*.doc -delete
* find . -type f -iname "\*.doc" -delete
* rm -f \*.doc \*\*/\*.doc
* find . -name "\*.doc" -delete
* find ./ -name '\*.doc' -delete
* find -name "\*.doc" -exec rm {} \;
* find -type f -name '\*.doc' -print -delete
* There is a file named access.log in the current working directory. Print all lines in this file that contains the string "GET".
* grep GET access.log
* cat access.log | grep GET
* grep "GET" access.log
* cat access.log | grep "GET"
* grep 'GET' access.log
* cat access.log | grep -e "GET"
* grep 'GET' ./access.log
* cat access.log | grep 'GET'
* cat access.log |grep GET
* grep GET \*
* grep -i "GET" access.log
* grep "GET" ./access.log
* cat access.log |grep "GET"
* cat access.log|grep GET
* cat access.log|grep "GET"
* cat access.log | grep -i "GET"
* grep GET ./access.log
* grep "GET" \*
* grep -w "GET" access.log
* grep -i get access.log
* grep GET access.log
* grep -i "get" access.log
* cat access.log| grep GET
* grep -F "GET" access.log
* grep -r "GET" access.log
* cat access.log | grep -i get
* grep -i GET access.log
* grep -e "GET" access.log
* cat access.log | grep -i GET
* cat access.log | grep -E "GET"
* grep "GET" "access.log"
* cat access.log| grep "GET"
* awk '/GET/ {print}' access.log
* cat access.log | grep -i "get"
* grep "GET" access.log
* grep GET access.log
* grep -F 'GET' access.log
* cat ./access.log | grep GET
* tail access.log | grep "GET"
* grep "GET" access.log
* cat access.log|grep 'GET'
* tail access.log | grep GET
* grep -e GET access.log
* cat access.log | grep -e GET
* cat access\* | grep GET
* grep -w GET access.log
* grep -i 'GET' access.log
* grep "GET" access\*
* grep -F GET access.log
* grep GET \*\*access.log
* Print all files in the current directory, one per line (not the path, just the filename) that contain the string "500".
* grep -l 500 \*
* grep -l "500" \*
* ls | grep -lR 500
* grep -l '500' \*
* ls | grep -lr 500
* grep -ls 500 \*
* ls | grep -lr "500"
* grep -rl "500"
* grep -lr 500
* grep -rl 500
* grep -lr "500"
* ls | grep -lR "500"
* grep 500 \* -l
* grep -rl "500" . | xargs -I {} basename {}
* grep -l "500" \* | xargs -n 1 basename
* ls | grep -l "500" \*
* grep 500 -l \*
* grep "500" \* -l
* grep -l "500" \* | xargs -I {} basename {}
* ls | grep -rl 500
* grep -ls "500" \*
* ls | grep -l 500 \*
* find . -type f -exec grep -l "500" {} \; | xargs -I {} basename {}
* grep -rl \* -e 500
* find . -type f -exec grep -l "500" {} \; | xargs -n 1 basename
* ls | grep -lr '500'
* grep 500 -lr
* grep "500" -l \*
* grep -l 500 \*
* grep -Ril "500"
* ls | grep -rl "500"
* find . -type f -exec grep -l "500" {} \; | xargs -I{} basename {}
* grep -rl '500'
* grep -rl "500" \*
* grep -lr "500" \*
* grep -lr '500'
* grep -lR 500
* grep -lR "500"
* grep 500 -rl
* grep -l 500 \*\*
* grep -rl "500" . | xargs -I{} basename {}
* cat access.log access.log.1 access.log.2 | grep -lr "500"
* grep -Rl 500
* grep -r -l 500
* grep -lr 500 \*
* find . | grep -rl 500
* grep -l "500" \* | xargs -I{} basename {}
* grep -Rl "500"
* grep -l -r 500
* grep -l "500" \* | sed 's/.\*\///'
* Print the relative file paths, one path per line for all filenames that start with "access.log" in the current directory.
* ls
* ls access.log\*
* find . -name "access.log\*"
* ls -r access.log\*
* ls | grep access.log
* find -name "access.log\*"
* find . -name 'access.log\*'
* find . -type f -name "access.log\*" -printf "%P\n"
* ls | grep "access.log"
* find . -name "access.log\*" -printf "%P\n"
* find . -name "access.log\*" -type f -printf "%P\n"
* find . -type f -name "access.log\*"
* ls -1 access.log\*
* find access.log\*
* ls \*
* ls | grep "access.log\*"
* find -name 'access.log\*'
* ls -a access.log\*
* find . -type f -name "access.log\*" -exec basename {} \;
* grep "access.log" | ls
* ls -1
* ls -r
* find . -type f -name 'access.log\*' -printf "%P\n"
* ls ./access.log\*
* ls | grep 'access.log'
* ls access\*
* ls "access.log"\*
* ls -d access.log\*
* grep access.log | ls
* ls | grep "^access.log"
* find . -name "access.log\*" -print
* find . -name "access.log\*" -type f
* ls -h
* find . -type f -name 'access.log\*'
* ls -p
* ls -F
* find ./ -name "access.log\*"
* find -type f -name "access.log\*"
* find . -name "access.log\*" -type f -exec basename {} \;
* find . -iname "access.log\*"
* find . -name 'access.log\*' -printf '%P\n'
* ls access.log\*
* ls | grep ^access.log
* ls -rt
* ls \*access.log\*
* find . -name 'access.log\*' -printf "%P\n"
* ls access.\*
* ls |grep access.log
* find . -name "\*access.log\*"
* ls -1 ./access.log\*
* Print all matching lines (without the filename or the file path) in all files under the current directory that start with "access.log" that contain the string "500".
* Note that there are no files named access.log in the current directory, you will need to search recursively.
* grep -r -h "500"
* grep -rh 500
* ls | grep -rh 500
* grep -r -h 500
* grep -rh "500"
* find . -name "access.log\*" | xargs grep -h 500
* find . -name "access.log\*" | grep -rh "500"
* grep -h 500 \*\*/access.log\*
* grep -rh 500 \*
* find . -name "access.log\*" -exec grep 500 {} \;
* grep -hr 500
* cat \*\*/access.log\* | grep 500
* grep -r -h '500'
* ls | grep -rh "500"
* grep -rh '500'
* grep -r -h 500 \*
* grep -hr "500"
* find . -name "access.log\*" -exec grep "500" {} \;
* grep -r -h 500 \*\*/access.log\*
* grep -h -r 500
* grep -rh 500 \*\*/access.log\*
* grep -r -h "500" --include="access.log\*" .
* cat \*\*/access.log\* | grep "500"
* grep -rh "500" \*
* find . -type f -name "access.log\*" -exec grep -h "500" {} \;
* grep -rh "500" --include="access.log\*" .
* find -name "access.log\*" | grep -rh "500"
* grep -h "500" \*\*/access.log\*
* grep -Rh 500
* find . -name "access.log\*" | grep -rh 500
* find . -name 'access.log\*' -exec grep 500 {} \;
* grep -h -r "500"
* ls | grep -rh 500 \*
* find . -name "access.log\*" | xargs grep -h "500"
* grep -h "500" $(find . -type f -name "access.log\*")
* find . -type f -name "access.log\*" -exec grep 500 {} \;
* grep -hr 500 \*
* find . -type f -name "access.log\*" -exec grep "500" {} \;
* grep -hr '500'
* grep -rh "500" \*\*/access.log\*
* grep -r -h "500" \*\*/access.log\*
* grep -Rh "500"
* grep -r -h '500' --include 'access.log\*'
* grep -h "500" $(find . -name "access.log\*")
* find . -name 'access.log\*' | xargs grep -h 500
* find . -name "access.log\*" | xargs grep "500" -h
* grep '500' -rh
* find . -name access.log\* | grep 500 -rh
* grep -r -h "500" --include "access.log\*"
* find -name "access.log\*" -exec grep 500 {} \;
* Extract all IP addresses from files that start with "access.log" printing one IP address per line.
* grep -ro ^[0-9.]\*
* find . -name "access.log\*" | xargs grep -Eo '^[^ ]+'
* cat \*\*/access.log\* | grep -ro ^[0-9.]\*
* awk '{print $1}' \*\*/access.log\*
* grep -ro ^[0-9.]\* .
* grep -rhEo "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" \*\*/access.log\*
* grep -or ^[0-9.]\*
* grep -oh ^[0-9.]\* \*\*/access.log\*
* grep -r . | awk '{print $1}'
* grep -roh ^[0-9.]\*
* grep -ro ^[0-9.]\* \*\*/access.log\*
* grep -r -h -o '[0-9]\*\.[0-9]\*\.[0-9]\*\.[0-9]\*' --include 'access.log\*'
* grep -r -o ^[0-9.]\*
* find -name access.log\* -exec awk '{print $1}' {} \;
* grep -ro "^[0-9.]\*"
* grep -ro ^[\*-9]\*
* awk '{ print $1 }' \*\*/access.log\*
* find . -name "access.log\*" -exec awk '{print $1}' {} \;
* grep -ro ^[0-9.]\*
* find -name "access.log\*" -type f | grep -ro ^[0-9.]\*
* grep -ro ^[\*-9.]\*
* grep -ro ^[0-9.]\*
* for i in $(find . -name "access.log\*"); do cat $i | awk '{print $1}'; done
* grep -ro '^[0-9.]\*'
* grep -o ^[0-9.]\* \*\*/access.log\*
* grep -rho ^[0-9.]\*
* grep -Ro ^[0-9.]\*
* cat \*\*/access.log\* | awk '{print $1}'
* find . -type f -name "access.log\*" -exec grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' {} \;
* grep -hr . | awk '{print $1}'
* grep -ro ^[0-9.]\* \*\*/access.log\*
* grep -Pho '^\d+\.\d+\.\d+\.\d+' \*\*/access.log\*
* grep -E ^[0-9.]\* -rho
* find . -name 'access.log\*' -exec awk '{print $1}' {} \;
* grep -roE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
* grep -rhEo "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}"
* find -name 'access.log\*' | grep -ro -E "([0-9]{1,3}[\.]){3}[0-9]{1,3}"
* grep -Goh ^[0-9.]\* \*\*/access.log\*
* find . -name "access.log\*" | grep -r "" | cut -d ':' -f 2 | cut -d ' ' -f 1
* grep -Eoh "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" \*\*/access.log\*
* grep -r \.|awk '{print $1}'
* grep -R -o -E "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" \* | cat
* find . -name "access.log\*" -exec egrep -ho "^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+" {} \;
* grep -rw . | awk '{print $1}'
* grep -roE "([0-9]{1,3}\.){3}[0-9]{1,3}"
* find -name "access.log\*" -exec egrep -o "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" {} \;
* find -name access.log\\* -exec awk '{print $1}' {} \;
* grep -oh '[0-9]\*\.[0-9]\*\.[0-9]\*\.[0-9]\*' \*\*/access.log\*
* find . -type f -name "access.log\*" -exec cut -d " " -f1 {} \;
* find . -type f -iname "access.log\*" -exec grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' {} \;
* Count the number of files in the current working directory. Print the number of files as a single integer.
* ls -l | wc -l
* find . -type f | wc -l
* find . -maxdepth 1 -type f | wc -l
* ls -A | wc -l
* find . -type f | wc --lines
* ls -l|wc -l
* ls -l |wc -l
* find -type f | wc -l
* ls -l| wc -l
* ls -la | grep -E ^'total [0-9]\*|\*[0-9] \.$|\*[0-9] \.\.$' -v | wc -l
* ls -R | wc -l
* ls -l | wc -l
* find -L . -type f | wc -l
* ls -1A | wc -l
* ls -s | wc -l
* find . -type f -print | wc -l
* find . -type f |wc -l
* ls -A |wc -l
* ls -A1 | wc -l
* ls -l | grep -v ^l | wc -l
* ls -l | wc --lines
* find -maxdepth 1 -type f|wc -l |awk '{print $1}'
* find . -type f|wc -l
* find ./ -type f | wc -l
* ls -A|wc -l
* find . -type f | wc -l
* find . -type f -maxdepth 1 | wc -l
* ls -s| wc -l
* find . -type f| wc -l
* find -type f | wc | awk -F' ' '{print $1}'
* ls -A| wc -l
* ls -l | wc -l
* ls -l | wc -l
* ls -ll |wc -l
* find -type f|wc -l
* ls -A | echo $(wc -l)
* find . ! -name . -prune -print | grep -c /
* ls -R . | wc -l
* find -type f -maxdepth 1 | wc -l
* ls -s1 | wc -l
* find -maxdepth 1 -type f | wc -l
* ls -A | wc -w
* ls -l |wc --lines
* ls -l | grep -v '^d' | wc -l
* ls -ll | wc -l
* ls -Aw1 | wc -l
* find -type f| wc -l
* find . -mindepth 1 | wc -l
* find . -type f -ls | wc -l
* ls -Ap | grep -v /$ | wc -l
* Print the contents of access.log sorted.
* sort access.log
* cat access.log | sort
* cat access.log|sort
* cat access.log |sort
* cat "access.log" | sort
* cat access.log | sort -u
* sort < access.log
* sort access.log | cat
* cat access.log| sort
* sort \*
* sort -u access.log
* ls | sort access.log
* cat | sort access.log
* find . -type f -name 'access.log\*' |xargs -I{} cat {} | sort
* sort access.log
* cat access.log | sort access.log
* sort "access.log"
* sort -d access.log
* cat access.log | sort -d
* sort a\*
* sort \*log
* sort access\*
* sort access.log\*
* sort <access.log
* sort < access.log
* cat access.log |sort -u
* sort ./access.log
* sort 'access.log'
* sort -b access.log
* sort -f access.log
* cat access.log|sort -u
* cat | sort access.log
* cat access.log | sort
* cat access.log | sort -b
* sort acc\*
* cat access.log | sort
* cat access.log | sort -i
* cat access.log |uniq|sort
* cat ./access.log | sort
* cat acces\* | sort
* cat access\* | sort
* sort -u < access.log
* sort access.log || cat
* sort < access.log | cat
* cat access.log | sort -s
* sort access.log && cat > access.log
* find . -name access.log | xargs cat | sort
* sort -M access.log
* sort \*\*/access.log\*
* tail access.log | sort
* Print the number of lines in access.log that contain the string "GET".
* grep GET access.log | wc -l
* grep "GET" access.log | wc -l
* grep -c "GET" access.log
* cat access.log | grep "GET" | wc -l
* grep -c GET access.log
* cat access.log | grep GET | wc -l
* cat access.log | grep -e "GET" | wc --lines
* grep 'GET' access.log | wc -l
* grep -c 'GET' access.log
* grep GET access.log|wc -l
* grep GET access.log |wc -l
* cat access.log | grep 'GET' | wc -l
* grep GET \* | wc -l
* grep -c 'GET' ./access.log
* grep GET access.log| wc -l
* cat access.log | grep "GET" | wc --lines
* grep "GET" access.log |wc -l
* cat access.log | grep -c "GET"
* cat access.log |grep GET |wc -l
* grep -c "GET" ./access.log
* grep "GET" access.log|wc -l
* cat access.log|grep GET|wc -l
* grep "GET" \* | wc -l
* grep -co GET \*
* grep -r "GET" | wc -l
* grep "GET" access.log| wc -l
* grep -i "GET" access.log | wc -l
* grep -r GET | wc -l
* cat access.log | grep -c GET
* cat access.log|grep "GET"|wc -l
* grep -o "GET" access.log | wc -l
* grep GET access.log | wc -l
* sort access.log | grep "GET" | wc -l
* cat access.log | grep -e "GET" | wc -l
* grep GET access.log | wc -l
* grep GET access.log | wc -l
* grep GET \* |wc -l
* grep -ro "GET" | wc -l
* grep -c GET ./access.log
* grep "GET" \* |wc -l
* grep -c GET access.log
* grep "GET" access.log | wc -l
* cat access.log | grep GET | wc --lines
* grep -wc GET access.log
* cat access.log | grep "GET" -c
* cat access.log| grep GET | wc -l
* grep -w "GET" access.log | wc -l
* grep -E "GET" access.log | wc -l
* cat access.log | grep -i GET | wc -l
* cat access.log | sort | grep GET | wc -l
* The file split-me.txt contains a list of numbers separated by a ; character.
* Split the numbers on the ; character, one number per line.
* cat split-me.txt | tr ";" "\n"
* tr ';' '\n' < split-me.txt
* cat split-me.txt | tr ';' '\n'
* sed "s/;/\n/g" split-me.txt
* sed 's/;/\n/g' split-me.txt
* cat split-me.txt | sed 's/;/\n/g'
* cat ./split-me.txt | sed s/\;/\\n/g
* tr ";" "\n" < split-me.txt
* cat \* | tr ';' '\n'
* cat split-me.txt | sed s/\;/\\n/g
* cat split-me.txt | tr ";" '\n'
* cat split-me.txt |tr ";" "\n"
* cat split-me.txt | sed "s/;/\n/g"
* cat split-me.txt|tr ";" "\n"
* cat split-me.txt | tr ';' "\n"
* cat split-me.txt| tr ";" "\n"
* tr ';' '\n' <split-me.txt
* sed -e 's/;/\n/g' split-me.txt
* cat split-me.txt|tr ';' '\n'
* cat split-me.txt | sed 's/\;/\n/g'
* for i in $(cat split-me.txt|sed 's/;/ /g');do echo $i;done
* sed 's/\;/\n/g' split-me.txt
* cat split-me.txt | sed -e 's/;/\n/g'
* cat split-me.txt |tr ';' '\n'
* cat \* | tr ";" "\n"
* cat split-me.txt| tr ';' '\n'
* tr ";" "\n" <split-me.txt
* cat split-me.txt|tr \; "\n"
* cat split-me.txt|sed 's/;/\n/g'
* cat split-me.txt | tr ";" "\n"
* cat split-me.txt | tr -s ";" "\n"
* cat \* | tr ';' '\n'
* tr ";" '\n' < split-me.txt
* cat split-me.txt |tr \; \\n
* cat split-me.txt |tr \; "\n"
* cat split-me.txt| tr \; "\n"
* sed 's/;/\n/g' < split-me.txt
* cat split-me.txt | tr ";", "\n"
* for i in $(cat split-me.txt | sed 's/;/ /g'); do echo $i; done
* <s\* tr ';' '\n'
* cat split-me.txt |tr ';' "\n"
* sed -e "s/;/\n/g" split-me.txt
* cat "split-me.txt" | tr ";" "\n"
* cat split-me.txt | sed "s/;/\n/g"
* cat split-me.txt | sed s/";"/"\n"/g
* cat split-me.txt | sed "s/;/""\n/g"
* cat split-me.txt | awk -F";" '{out=""; for(i=1;i<=NF;i++){out=out$i"\n"}; print out}'
* tr \; \\n < split-me.txt
* tr ';' "\n" < split-me.txt
* sed -e "s/;/\\n/g" split-me.txt
* Print the numbers 1 to 100 separated by spaces.
* echo {1..100}
* for i in {1..100}; do echo -n "$i "; done
* seq -s " " 100
* seq -s " " 1 100
* echo $(seq 1 100)
* seq -s ' ' 1 100
* seq 1 100 | tr '\n' ' '
* for ((i=1; i<=100; i++)); do echo -n "$i "; done
* echo {1..100} " "
* seq 1 100 | tr "\n" " "
* seq -s ' ' 100
* printf "%d " {1..100}
* for i in {1..100}; do echo -n $i" "; done
* for i in {1..100};do echo -n "$i ";done
* for (( i=1; i<=99; i++ ))
* do
* printf "$i "
* done
* echo 100
* for i in {1..100}; do printf "$i "; done
* echo `seq 1 100`
* printf '%s ' {1..100}
* seq 1 100 | xargs
* for i in {1..100}; do echo -n "$i " ; done
* seq 100 | tr '\n' ' '
* echo {1..100};
* echo {1..100}
* echo {1..100}\
* echo {1..100}""
* echo {1..100} ""
* seq 100 | tr "\n" " "
* for i in {1..100}; do echo -n "$i "; done;
* seq -s' ' 1 100
* echo "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100"
* echo {1..100} | tr ' ' ' '
* for i in {1..100};do echo -n "$i "; done
* echo -n {1..100}
* seq -s" " 1 100
* for i in $(seq 1 100); do echo -n "$i "; done
* seq 1 100 | xargs echo
* seq -s' ' 100
* for ((i=1; i<=100; i++)); do
  + echo -n "$i "
* done
* for ((i=1; i<=100; i++)); do
  + echo -n "$i "
* done
* for ((i=1; i<=100; i++)); do
  + echo -n "$i "
* done
* echo
* seq -s" " 100
* for ((i=1; i<=100; i++)); do
  + echo -n "$i "
* done
* echo
* seq 100 | xargs
* seq 1 100 | paste -s -d ' '
* for i in {1..100}; do printf $i" "; done
* seq -s " " 1 1 100
* seq --separator=" " 100
* for i in {1..100}; do echo -n $i" " ;done
* for((i=1;i<=100;i++)); do echo -n "$i "; done
* seq -s\ 1 100
* This challenge has text files (with a .txt extension) that contain the phrase "challenges are difficult". Delete this phrase from all text files recursively.
* Note that some files are in subdirectories so you will need to search for them.
* find . -name '\*.txt' -type f -exec sed -i '/challenges are difficult/d' {} \;
* sed -i 'challenge are difficult/d' \*\*/\*.txt
* sed -i "challenges are difficult" \*\*/\*.txt
* rm -r \*\*/\*.txt
* find . -type f -name "\*.txt" -exec sed -i 's/challenges are difficult//g' {} +
* sed -i 'challenges are difficult/d' \*\*/\*.txt
* sed -i "challenges are difficult/d" \*\*/\*.txt
* sed -i 's/challenges are difficult//g' \*\*/\*.txt
* sed -i 'challenges are difficult' \*\*/\*.txt
* find . -name "\*.txt" -exec sed -i 's/challenges are difficult//g' {} +
* find . -type f -name "\*.txt" -exec sed -i 's/challenges are difficult//g' {} \;
* sed -i '/challenges are difficult/d' \*\*/\*.txt
* find . -type f -name "\*.txt" -exec sed -i '/challenges are difficult/d' {} \;
* find . -name "\*.txt" -exec sed -i 's/challenges are difficult//g' {} \;
* sed -i "s/challenges are difficult//g" \*\*/\*.txt
* rm -rf \*\*/\*.txt
* find . -type f -name '\*.txt' -exec sed -i 's/challenges are difficult//g' {} +
* sed -i 'challenge are difficult' \*\*/\*.txt
* find . -type f -name "\*.txt" -exec sed -i '/challenges are difficult/d' {} +
* sed -i "challenge are difficult" \*\*/\*.txt
* find . -name "\*.txt" -type f -exec sed -i '/challenges are difficult/d' {} \;
* find . -name "\*.txt" -exec sed -i "s/challenges are difficult//g" {} +
* find . -name "\*.txt" -type f -exec sed -i 's/challenges are difficult//g' {} +
* sed -i "challenges are difficult\d" \*\*/\*.txt
* sed -i "/challenges are difficult/d" \*\*/\*.txt
* find . -type f -name '\*.txt' -exec sed -i 's/challenges are difficult//g' {} \;
* find . -name '\*.txt' -exec sed -i 's/challenges are difficult//g' {} \;
* find . -type f -name '\*.txt' -exec sed -i '/challenges are difficult/d' {} \;
* find . -type f -name "\*.txt" -exec sed -i "/challenges are difficult/d" {} \;
* find . -type f -name "\*.txt" -exec sed -i "s/challenges are difficult//g" {} \;
* find . -name "\*.txt" -type f -exec rm {} +
* sed -i "challanges are difficult" \*\*/\*.txt
* sed -i "challenge are difficult/d" \*\*/\*.txt
* find . -name "\*.txt" -exec sed -i '/challenges are difficult/d' {} \;
* sed -i "challendes are difficult" \*\*/\*.txt
* find . -name '\*.txt' -exec sed -i 's/challenges are difficult//g' {} +
* sed -i "challenges are difficult" "" \*\*/\*.txt
* find . -name "\*.txt" -type f -exec sed -i "/challenges are difficult/d" {} \;
* find . -name "\*.txt" -type f -exec sed -i 's/challenges are difficult//g' {} \;
* find ./ -iname "\*.txt" -print0 | xargs -0 sed -i 's/challenges are difficult//g'
* sed -i challenge are difficult \*\*/\*.txt
* sed -i 'chalenges are difficut/d' \*\*/\*.txt
* sed -i 'challnge are difficult/d' \*\*/\*.txt
* sed -i 'challenges are difficult/d' \*\*/\*.txt
* find -name "\*.txt" -exec sed -i '/challenges are difficult/d' {} \;
* find . -name "\*.txt" -exec sed -i "/challenges are difficult/d" {} \;
* sed -i "challenges are difficult " \*\*/\*.txt
* sed -i 'chalenges are difficult/d' \*\*/\*.txt
* find -name "\*.txt" -exec sed -i 's/challenges are difficult//g' {} \;
* find . -name "\*.txt" -exec sed -i "s/challenges are difficult//g" {} \;