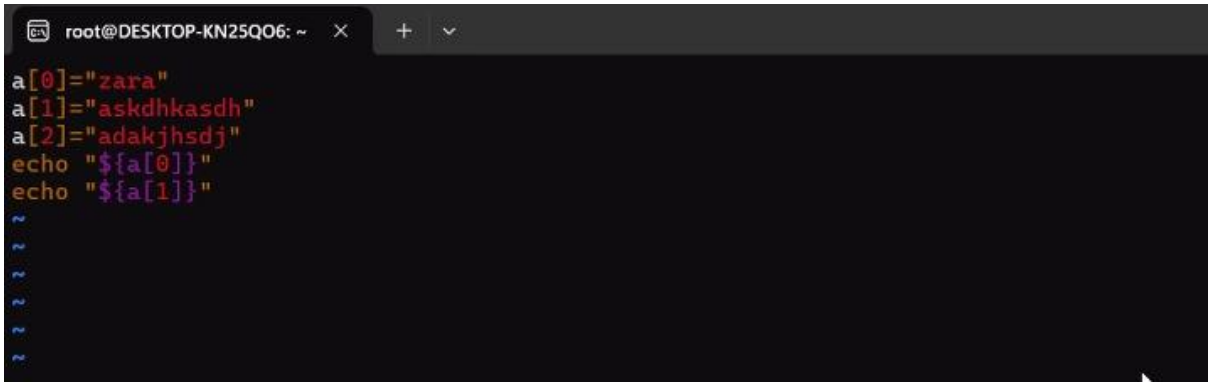
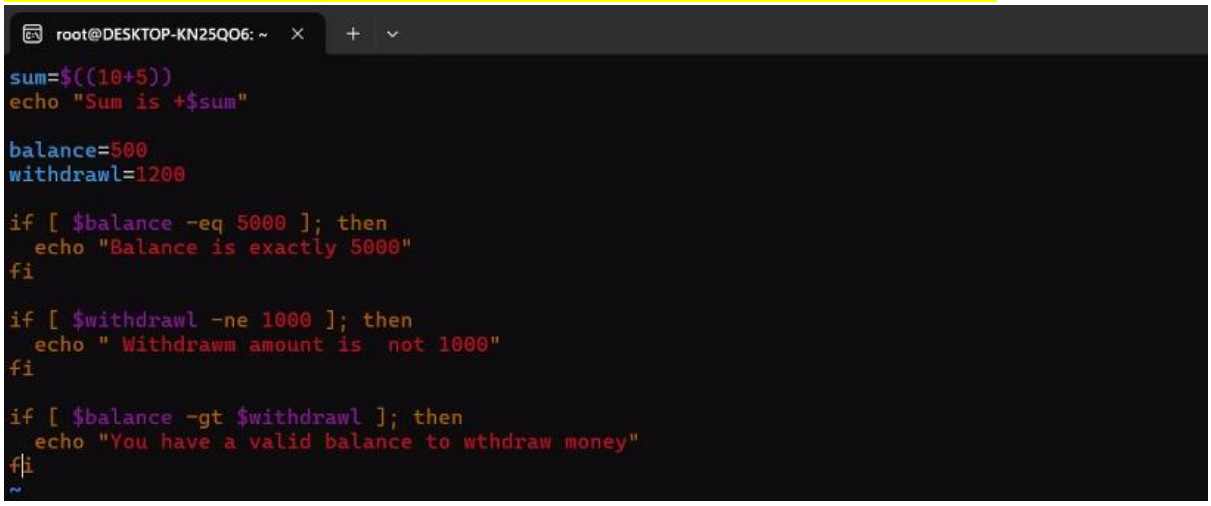
LINUX LECTURE-3 (day-8)

How to declare an array using indexes:

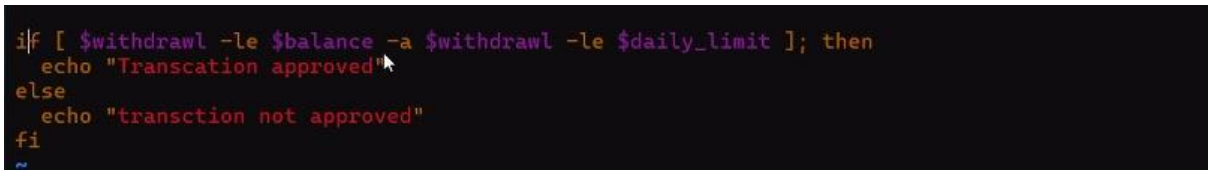


Operators and equal and not equal to operators in shell:

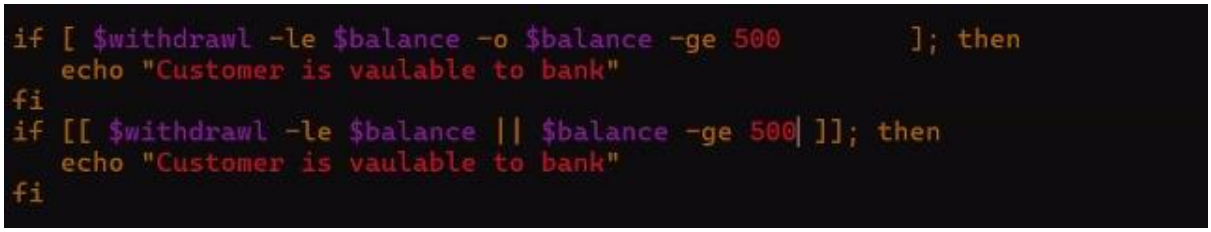


Logical operators:

Logical and (-a):



Logical or (-o or ||):



Logical Not (!):

```
if [[ ! $withdrawl -le $balance || $balance -ge 500 ]]; then
    echo "Customer is vaulable to bank"
fi
```

Want to Compare two strings:

```
if [ "$account_type" = "savings" ]; then
   echo "These is a saaving account"
fi
```

In shell scripting, `-n` and `-z` are used within conditional expressions (often with `if` statements) to check the length of a string. Here's a breakdown of their differences:

-n string

- Returns true (a zero exit status) if the length of *string* is greater than zero.
- In simpler terms, it checks if the string is *not* empty.

-z string

- Returns true if the length of *string* is zero.
- It checks if the string is empty.

```
if [ -z "$description" ]; then
    echo "deciption is not provided"
fi
```

To check if a file exists we use -e:

```
array_file="array.sh"
if [ -e $array_file ]; then
  echo " file exists"
fi
:wq
```

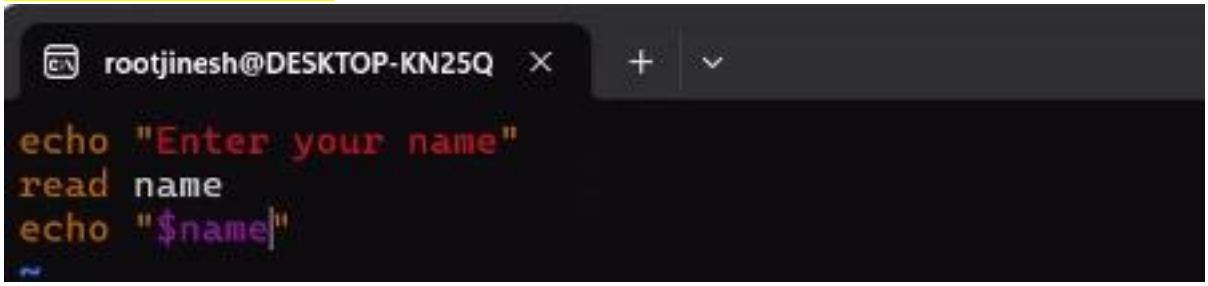
To check if directory exists or not -d:

```
if [ -d $dirt ]; then
  echo " directory exists"
fi:
```

To check for read write execute permission we can use -r -w -x:

```
filen="alpha.txt"
if [ -w $filen ]; then
   echo " have wirte permission"
else
   echo " no write permission"
fi
```

How to take input:



if u want to generate message with prompt

```
rootjinesh@DESKTOP-KN25QO6:~$ vi input.sh
rootjinesh@DESKTOP-KN25QO6:~$ ./input.sh
Enter your name
jinesh
jinesh
 Enter account number and password:121143
rootjinesh@DESKTOP-KN25QO6:~$ ./input.sh
```

```
echo "Enter your name"
read name
echo "$name"

read -p " Enter account number and password:" acn password
echo $acn
echo $password
```

we can use space to differentiate between two values when entering data.

To create a secretive password (-s stands for silent):

```
#echo "enter senstive password"
read -s -p " Enter passeord" p
```

so we won't see password when user enters it.

if we want to generate a timer for entering data here for 5 second:

```
read -t 5 -p "quick 5 sec" pin
echo "Enter your name"
read name
echo "$name"

read -p " Enter account number and password:" acn password
echo $acn
echo $password
#echo "enter senstive password"
read -s -p " Enter passeord" p
#echo $p
```

Switch case using entered value:

```
rootjinesh@DESKTOP-KN25QO6:~$ chmod 777 case.sh
rootjinesh@DESKTOP-KN25QO6:~$ ./case.sh
Enter selection [1-3]2
you have selected saving
rootjinesh@DESKTOP-KN25QO6:~$ ./case.sh
Enter selection [1-3]4
ramdam selection
rootjinesh@DESKTOP-KN25QO6:~$
```

rootjinesh@DESKTOP-KN25Q

```
read -p "Enter selection [1-3]" selection
case $selection in
  1) accounttype="checking"; echo " you have sleected checking";;
  2) accountype="saving"; echo "you have selected saving";;
  3) accountype="current"; echo " you ahev selected curemt";;
  *) accountype="random"; echo "ramdam selection";;
esac
```

REGULAR EXPRESSION:

Regular expressions (regex) are a powerful tool for pattern matching in strings. They are widely used in scripting languages for tasks like:

- **Metacharacters:** These have special meanings:
  - `.` (dot): Matches any single character except a newline.
  - `*` (asterisk): Matches the preceding character zero or more times.
  - `+` (plus): Matches the preceding character one or more times.
  - `?` (question mark): Matches the preceding character zero or one time.
  - `[]` (square brackets): Matches any single character within the brackets. `[aeiou]` matches any vowel. `[0-9]` matches any digit.
  - `()` (parentheses): Groups characters together. Also used for capturing matched groups.
  - `|` (pipe): Acts as an "OR" operator. `a|b` matches either "a" or "b".
  - `^` (caret): Matches the beginning of a line (or string, depending on the context).
  - `$` (dollar sign): Matches the end of a line (or string).
  - `\` (backslash): Escapes metacharacters, allowing you to match them literally. `\.` matches a literal dot.

Examples:

- `^Hello`: Matches strings that start with "Hello".
- `world$`: Matches strings that end with "world".
- `[0-9]+`: Matches one or more digits.
- `[a-zA-Z]*`: Matches zero or more letters (uppercase or lowercase).
- `[a-z]{5}`: Matches exactly five lowercase letters.
- `\d{3}-\d{3}-\d{4}`: Matches a phone number format like "123-456-7890".

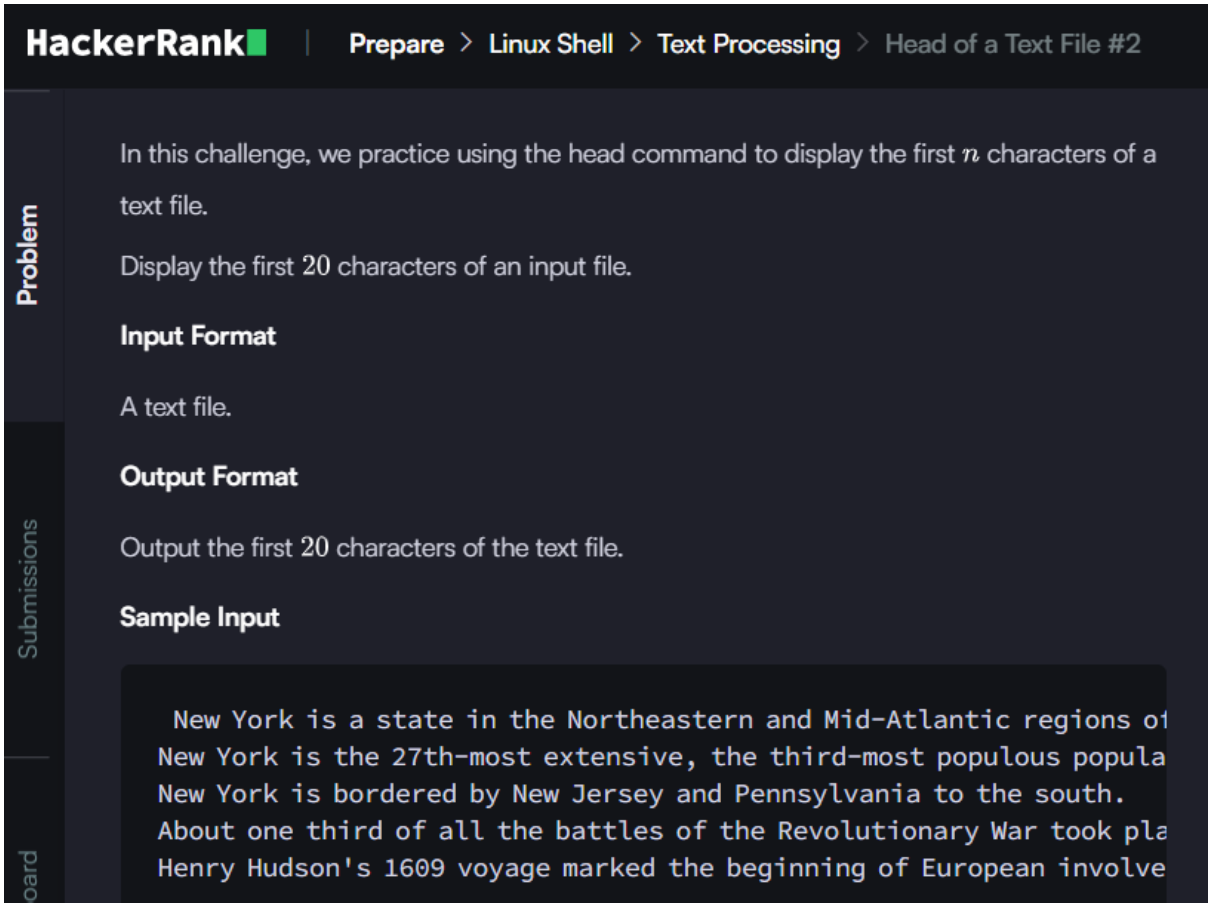
```
rootjinesh@DESKTOP-KN25QO6:~$ grep "^read" case.sh
read -p "Enter selection [1-3]" selection
rootjinesh@DESKTOP-KN25QO6:~$ vi case.sh
rootjinesh@DESKTOP-KN25QO6:~$ grep "^case" case.sh
case $selection in
rootjinesh@DESKTOP-KN25QO6:~$
```

```
rootjinesh@DESKTOP-KN25QO6:~$ grep "selection$" case.sh
read -p "Enter selection [1-3]" selection
```

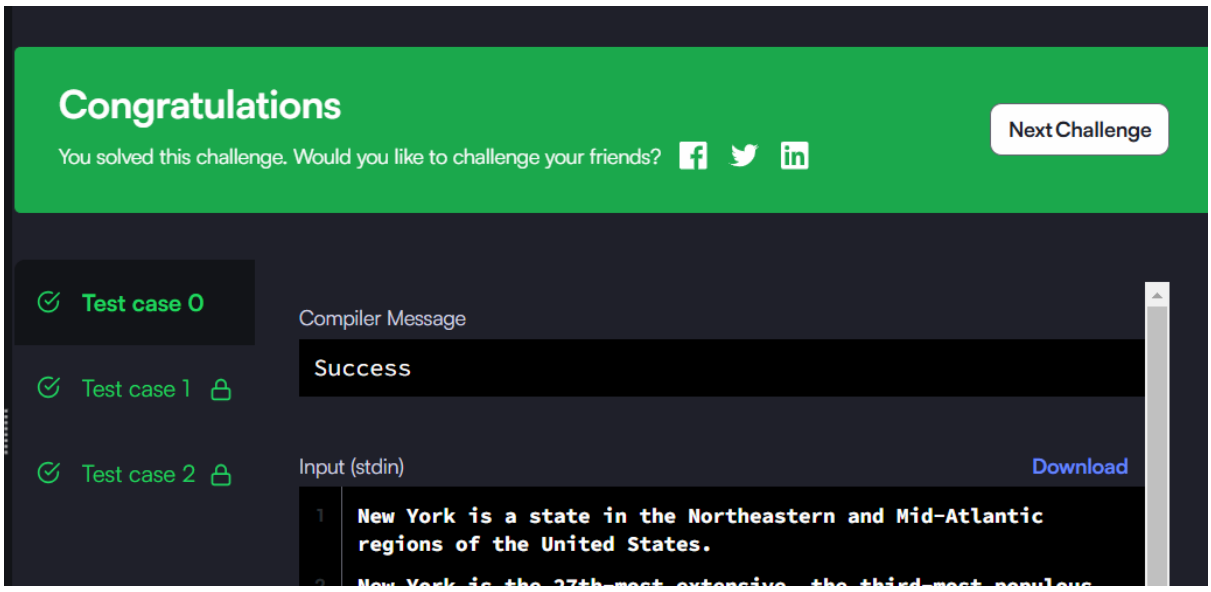
```
rootjinesh@DESKTOP-KN25QO6:~$ grep "s.lection$" case.sh
read -p "Enter selection [1-3]" selection
rootjinesh@DESKTOP-KN25QO6:~$ grep "[0-9]" case.sh
read -p "Enter selection [1-3]" selection
  1) accounttype="checking"; echo " you have sleected checking";;
  2) accountype="saving"; echo "you have selected saving";;
  3) accountype="current"; echo " you ahev selected curemt";;
rootjinesh@DESKTOP-KN25QO6:~$
```

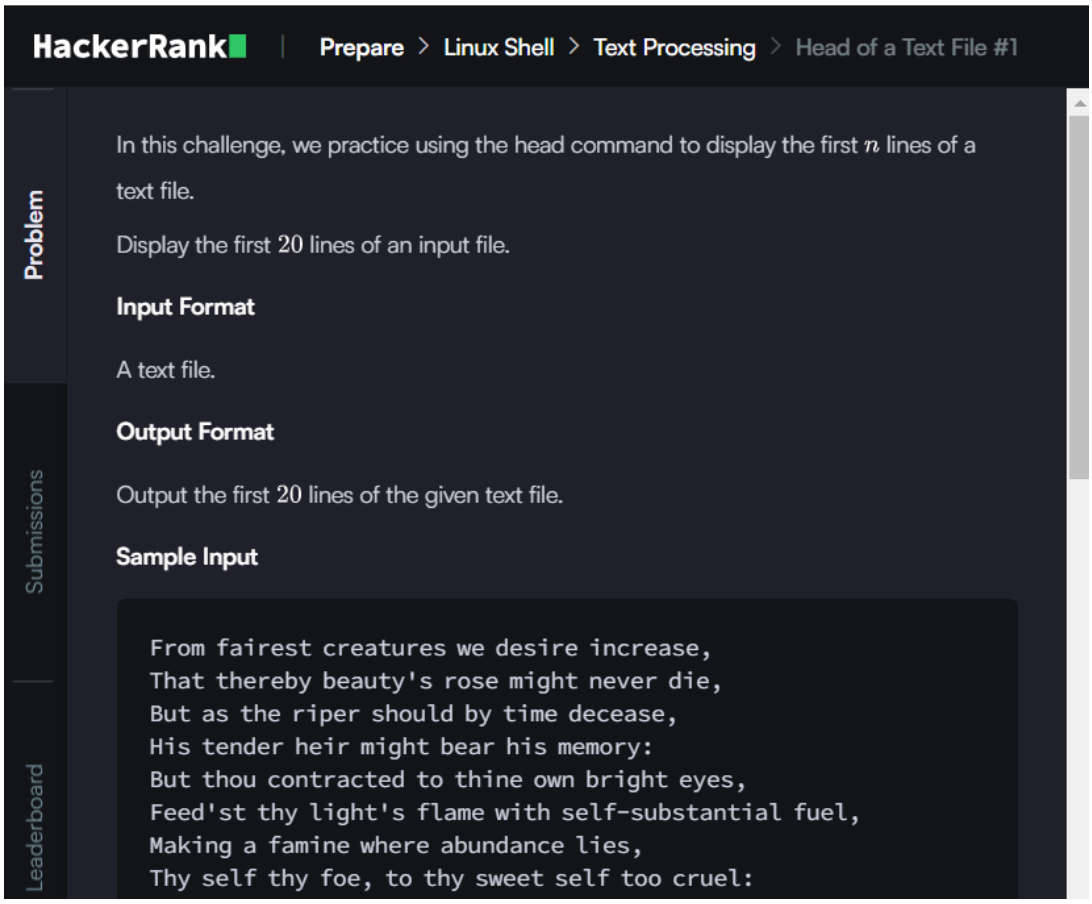
```
rootjinesh@DESKTOP-KN25QO6:~$ grep "[a-zA-Z]" case.sh
read -p "Enter selection [1-3]" selection
case $selection in
  1) accounttype="checking"; echo " you have sleected checking";;
  2) accountype="saving"; echo "you have selected saving";;
  3) accountype="current"; echo " you ahev selected curemt";;
  *) accountype="random"; echo "ramdam selection";;
esac
rootjinesh@DESKTOP-KN25QO6:~$
```

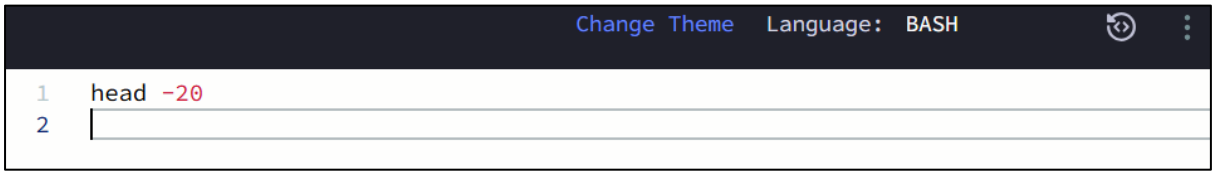
```
rootjinesh@DESKTOP-KN25QO6:~$ grep "s*n" case.sh
read -p "Enter selection [1-3]" selection
case $selection in
  1) accounttype="checking"; echo " you have sleected checking";;
  2) accountype="saving"; echo "you have selected saving";;
  3) accountype="current"; echo " you ahev selected curemt";;
  *) accountype="random"; echo "ramdam selection";;
rootjinesh@DESKTOP-KN25QO6:~$ grep "s*on" case.sh
read -p "Enter selection [1-3]" selection
case $selection in
  *) accountype="random"; echo "ramdam selection";;
rootjinesh@DESKTOP-KN25QO6:~$
```

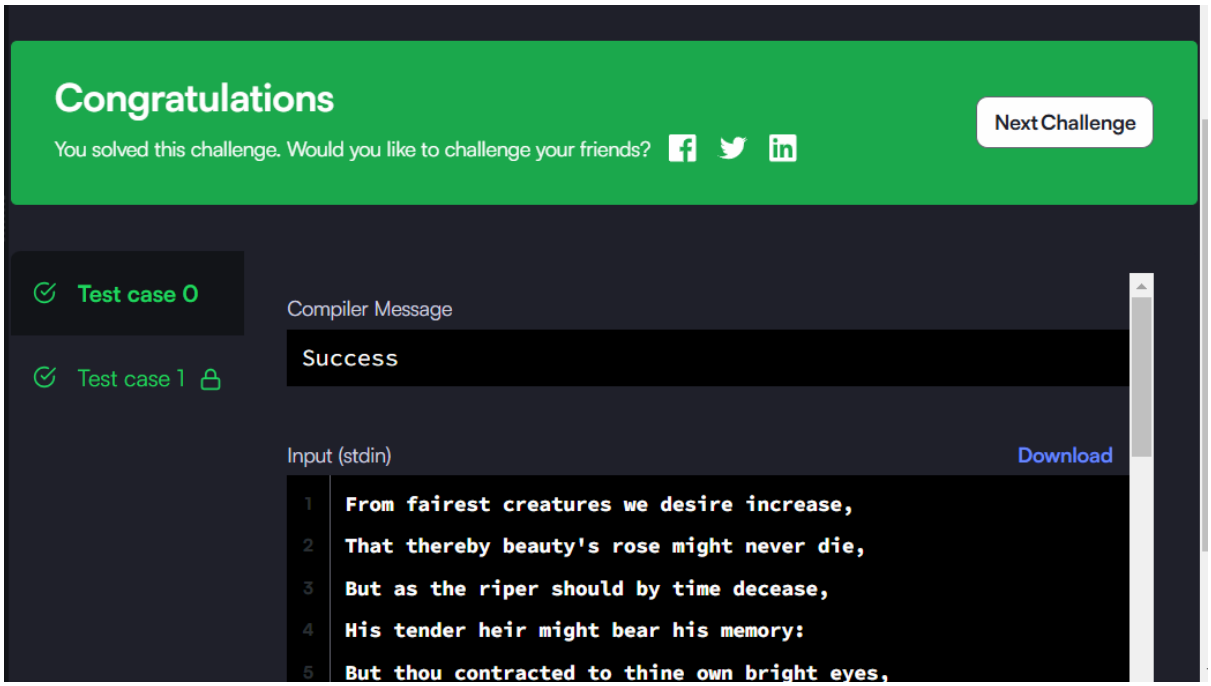

HackerRank
Prepare > Linux Shell > Text Processing > Head of a Text File #2
Problem
Submissions
In this challenge, we practice using the head command to display the first n characters of a text file.
Display the first 20 characters of an input file.
Input Format
A text file.
Output Format
Output the first 20 characters of the text file.
Sample Input
New York is a state in the Northeastern and Mid-Atlantic regions of
New York is the 27th-most extensive, the third-most populous popula
New York is bordered by New Jersey and Pennsylvania to the south.
About one third of all the battles of the Revolutionary War took pla
Henry Hudson's 1609 voyage marked the beginning of European involve

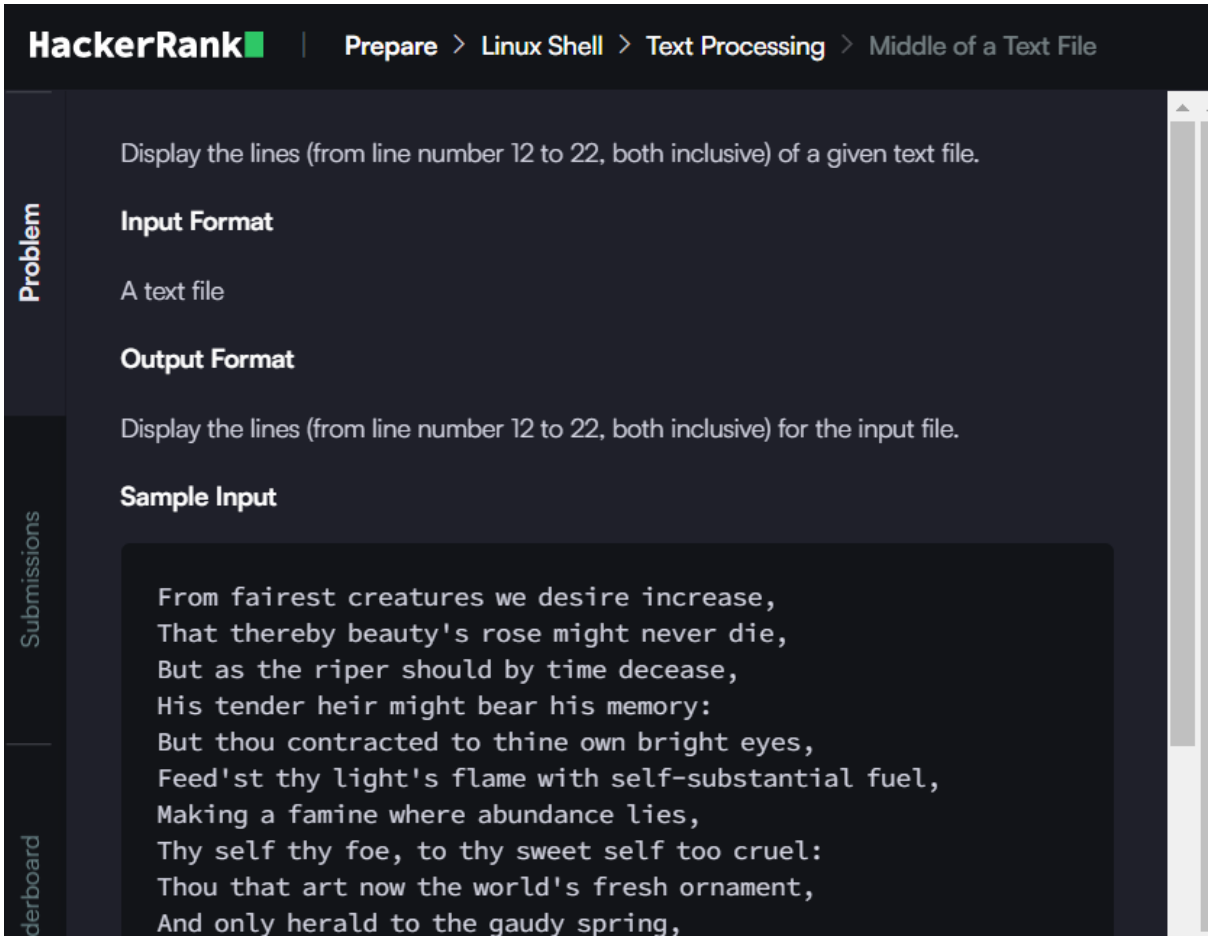

Change Theme
Language: BASH
1
head -c 20

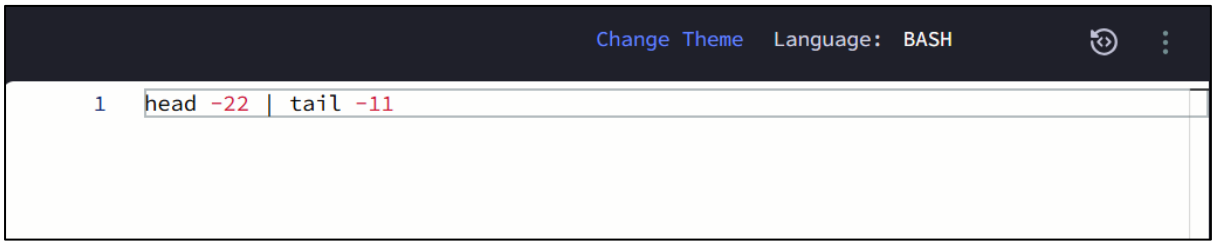

Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Compiler Message
Success
Input (stdin)
Download
New York is a state in the Northeastern and Mid-Atlantic regions of the United States.

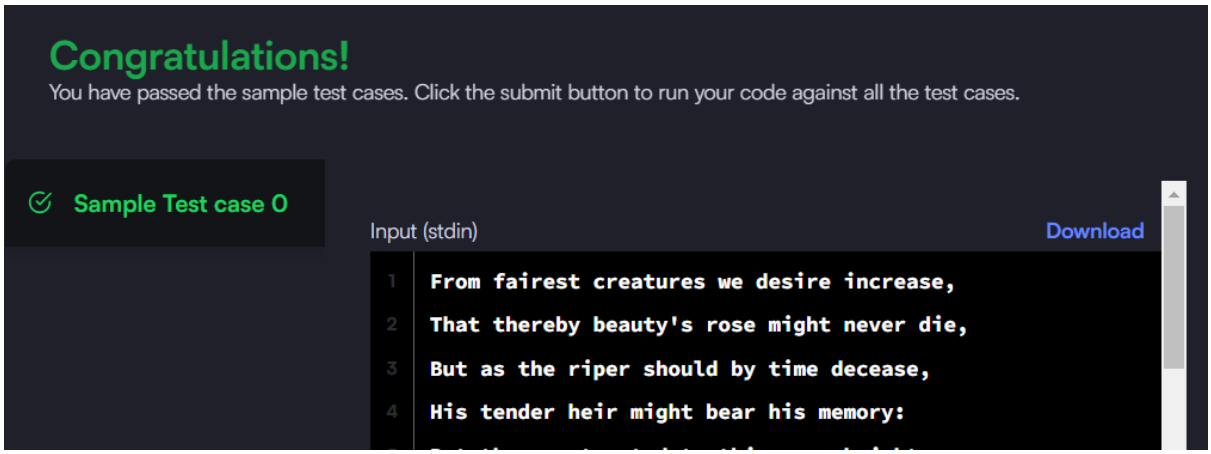

HackerRank
Prepare > Linux Shell > Text Processing > Head of a Text File #1
Problem
Submissions
Leaderboard
In this challenge, we practice using the head command to display the first n lines of a text file.
Display the first 20 lines of an input file.
Input Format
A text file.
Output Format
Output the first 20 lines of the given text file.
Sample Input
From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thy self thy foe, to thy sweet self too cruel:

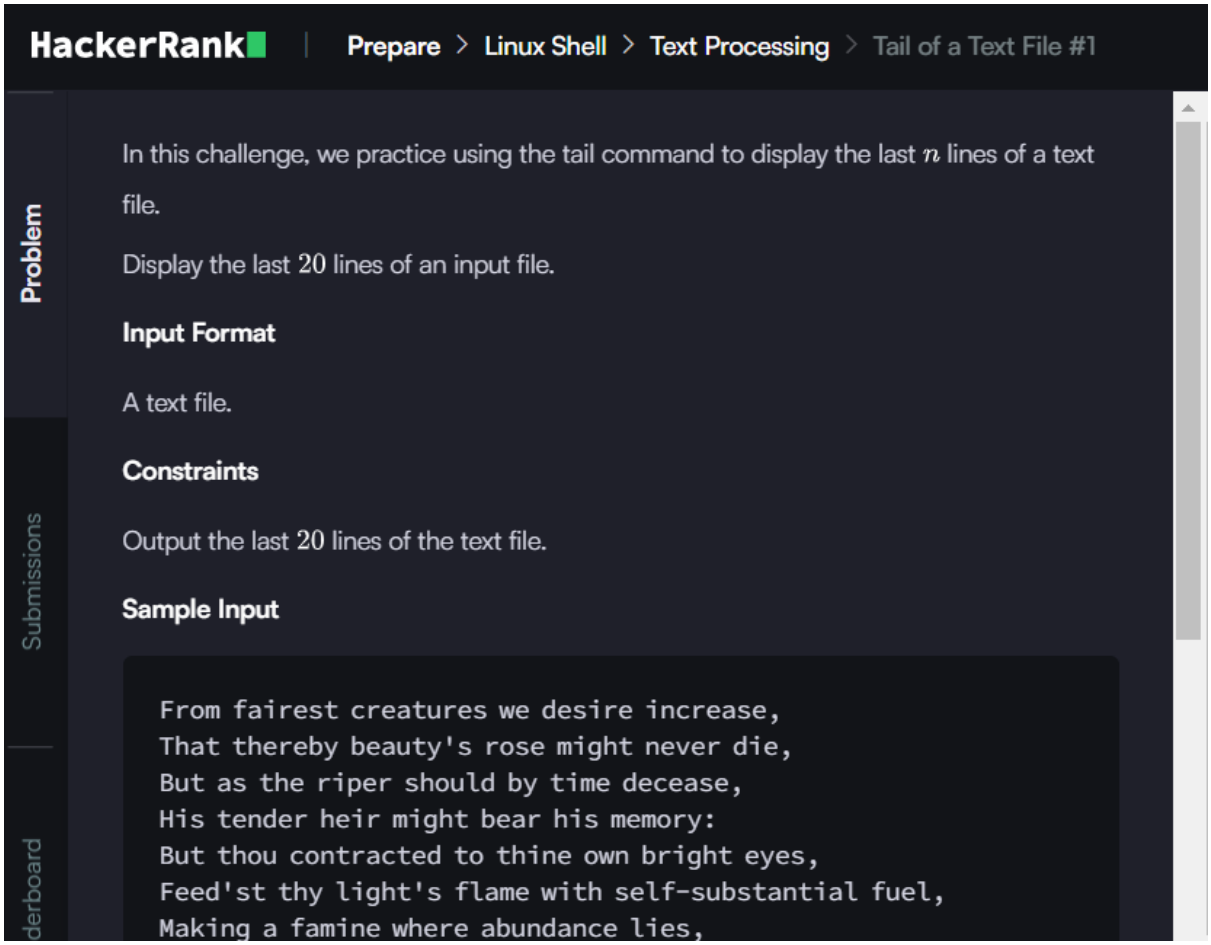

Change Theme
Language: BASH
1 head -20
2


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success
Input (stdin)
Download
From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou contracted to thine own bright eyes,

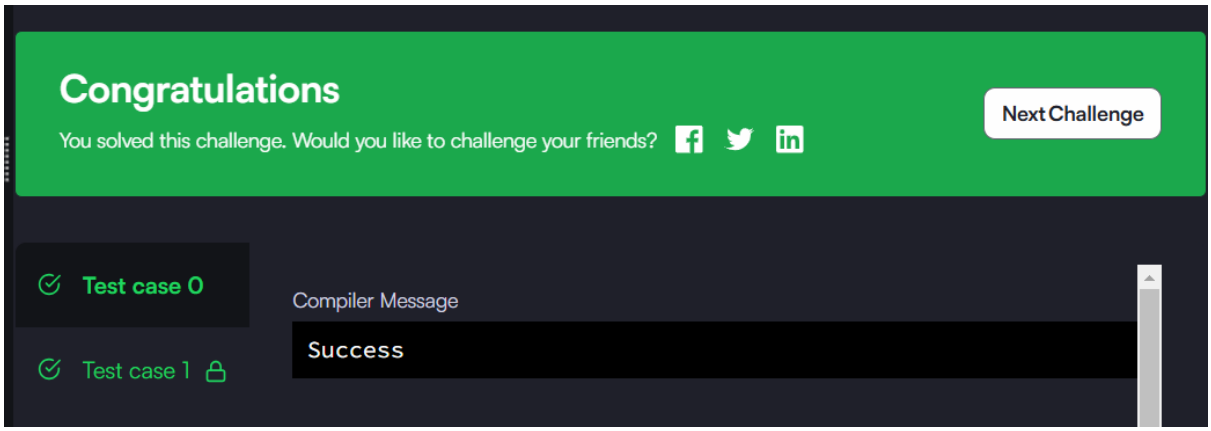

HackerRank
Prepare > Linux Shell > Text Processing > Middle of a Text File
Problem
Submissions
derboard
Display the lines (from line number 12 to 22, both inclusive) of a given text file.
Input Format
A text file
Output Format
Display the lines (from line number 12 to 22, both inclusive) for the input file.
Sample Input
From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thy self thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,

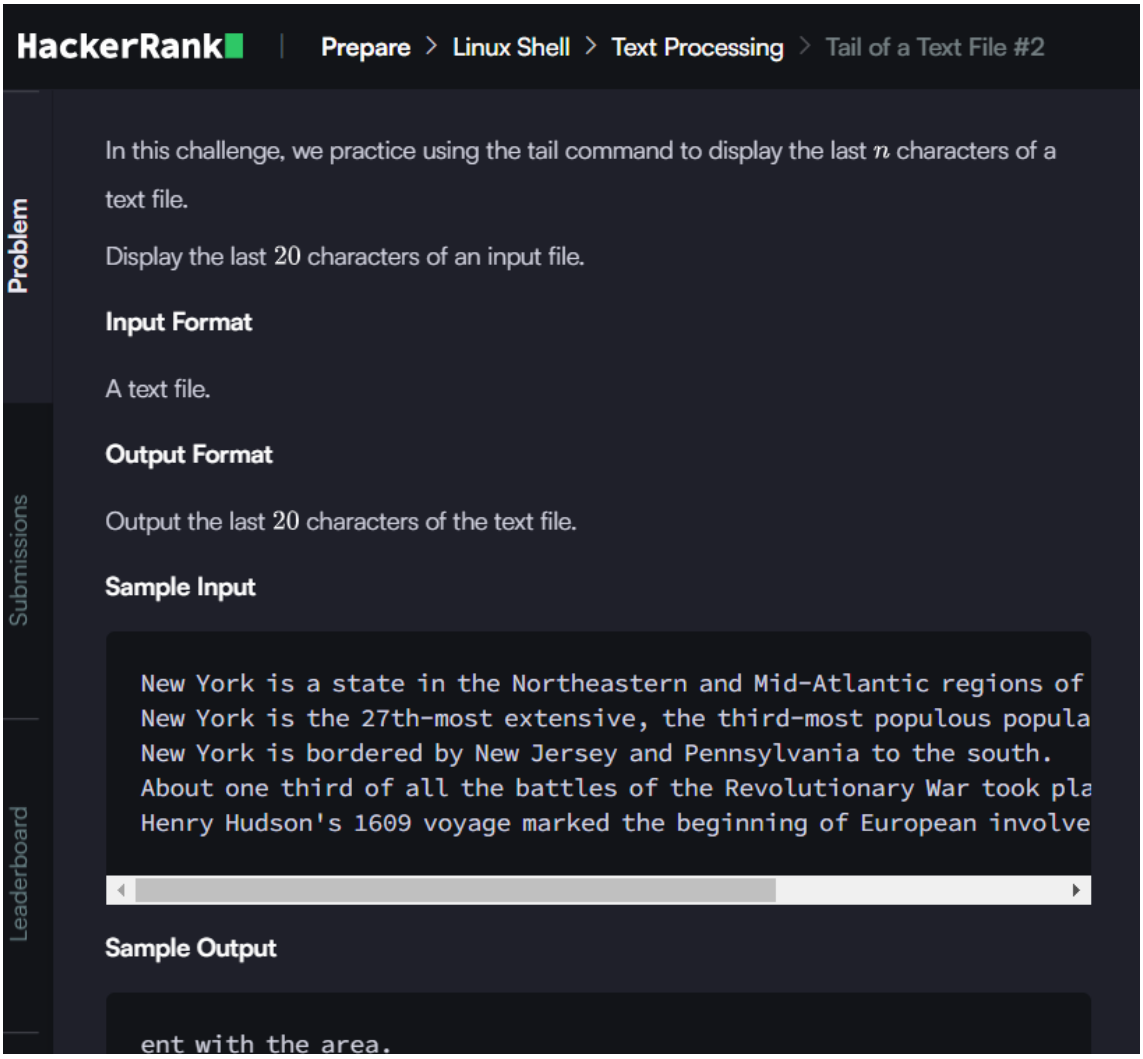

Change Theme
Language: BASH
1
head -22 | tail -11

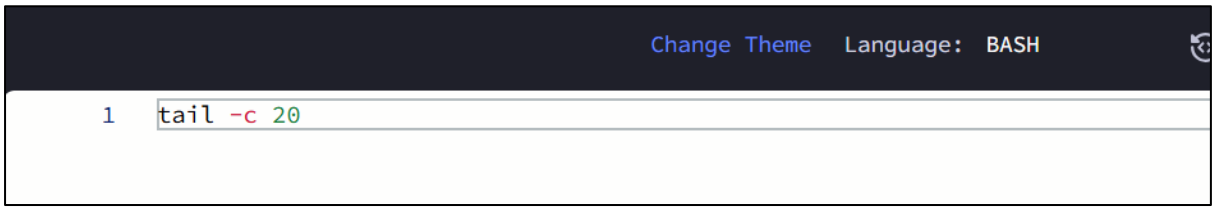

Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
1 From fairest creatures we desire increase,
2 That thereby beauty's rose might never die,
3 But as the riper should by time decease,
4 His tender heir might bear his memory:

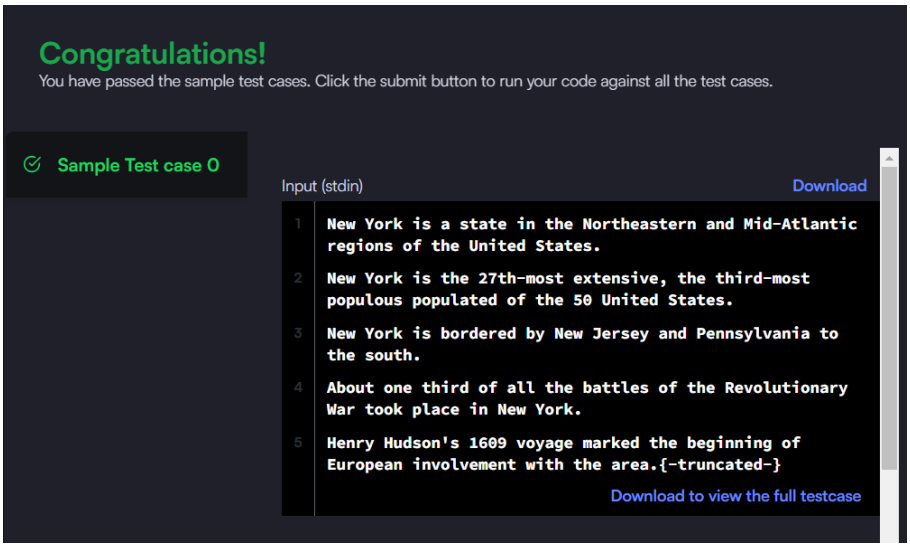

HackerRank
Prepare > Linux Shell > Text Processing > Tail of a Text File #1
Problem
Submissions
derboard
In this challenge, we practice using the tail command to display the last n lines of a text file.
Display the last 20 lines of an input file.
Input Format
A text file.
Constraints
Output the last 20 lines of the text file.
Sample Input
From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,


Change Theme
Language: BASH
1
tail -20


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success


HackerRank
Prepare > Linux Shell > Text Processing > Tail of a Text File #2
Problem
Submissions
Leaderboard
In this challenge, we practice using the tail command to display the last n characters of a text file.
Display the last 20 characters of an input file.
Input Format
A text file.
Output Format
Output the last 20 characters of the text file.
Sample Input
New York is a state in the Northeastern and Mid-Atlantic regions of
New York is the 27th-most extensive, the third-most populous popula
New York is bordered by New Jersey and Pennsylvania to the south.
About one third of all the battles of the Revolutionary War took pla
Henry Hudson's 1609 voyage marked the beginning of European involve
Sample Output
ent with the area.


Change Theme
Language: BASH
1
tail -c 20


Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
1 New York is a state in the Northeastern and Mid-Atlantic regions of the United States.
2 New York is the 27th-most extensive, the third-most populous populated of the 50 United States.
3 New York is bordered by New Jersey and Pennsylvania to the south.
4 About one third of all the battles of the Revolutionary War took place in New York.
5 Henry Hudson's 1609 voyage marked the beginning of European involvement with the area.{-truncated-}
Download to view the full testcase

HackerRank | Prepare > Linux Shell > Text Processing > 'Tr' Command #1

Problem | Submissions | Leaderboard

In this challenge, we practice using the tr command because it is a useful translation tool in Linux.

In a given fragment of text, replace all parentheses ( ) with box brackets [ ].

Input Format

A block of ASCII text.

Output Format

Output the text with all parentheses ( ) replaced with box brackets [ ].

Sample Input

```
int i=(int)5.8
(23 + 5)*2
```

Sample Output

```
int i=[int]5.8
[23 + 5]*2
```

Change Theme Languag

```
1  tr '()' '[]'
```

Congratulations

You solved this challenge. Would you like to challenge your friends?

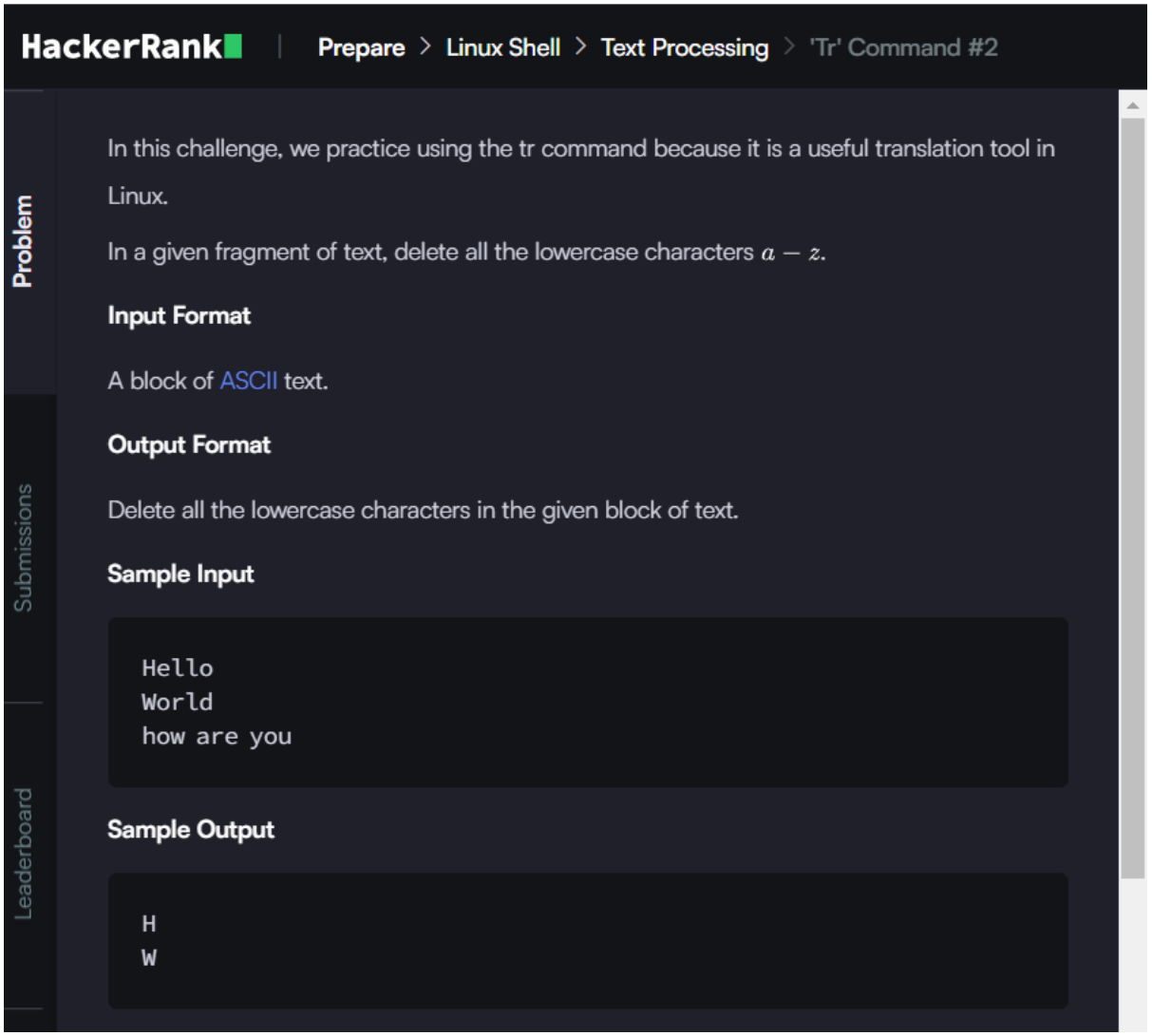
Next Challenge

Test case 0

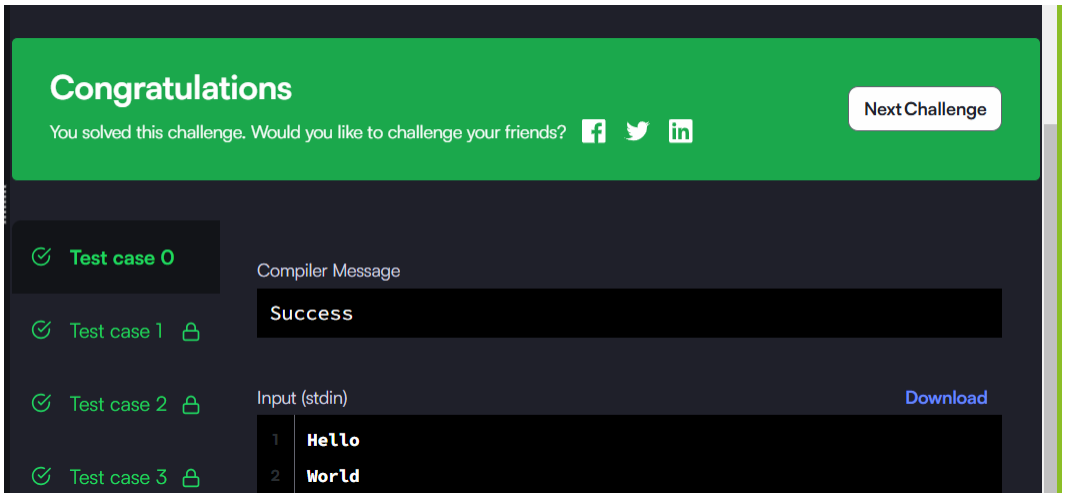
Test case 1

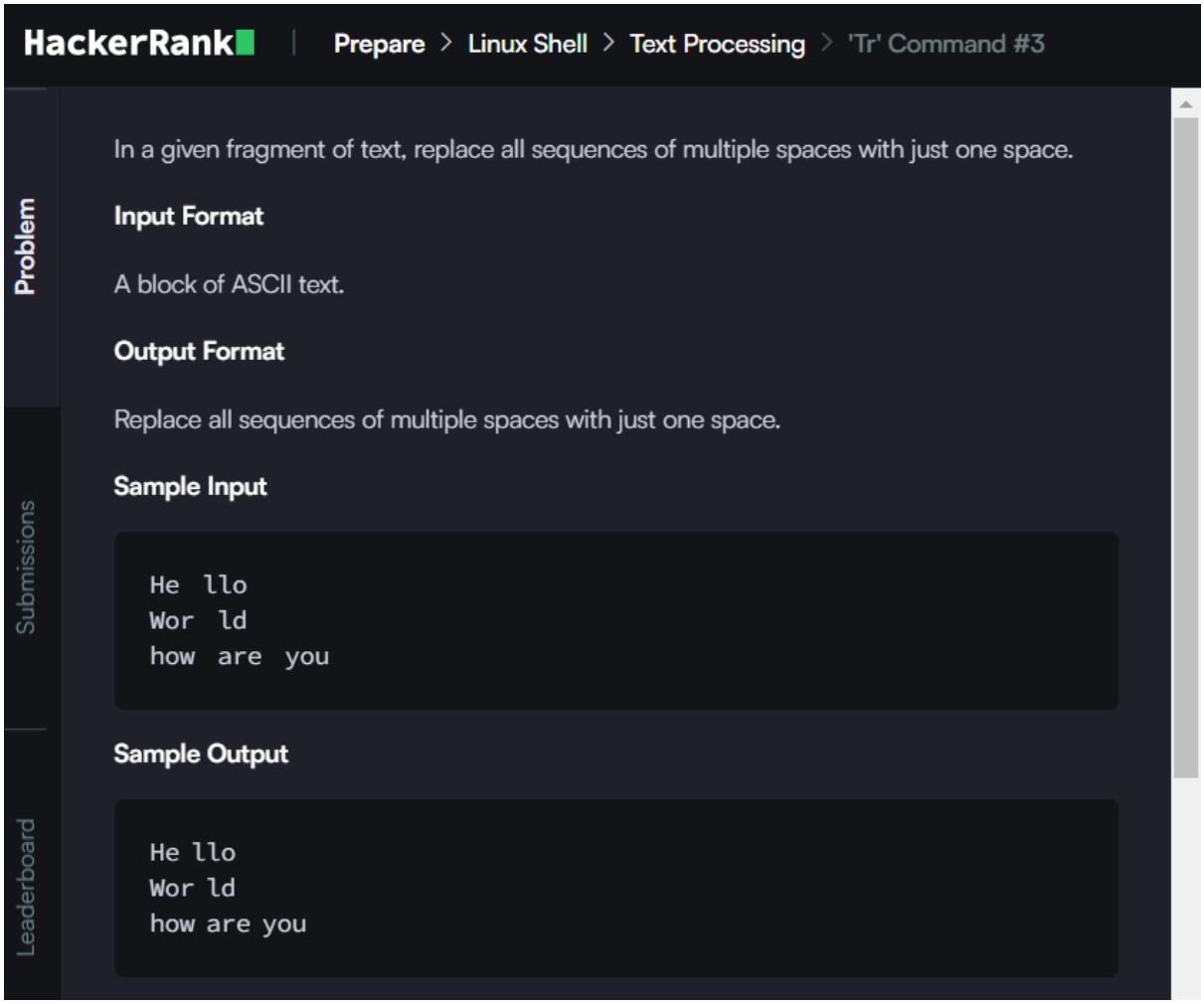
Compiler Message

Success

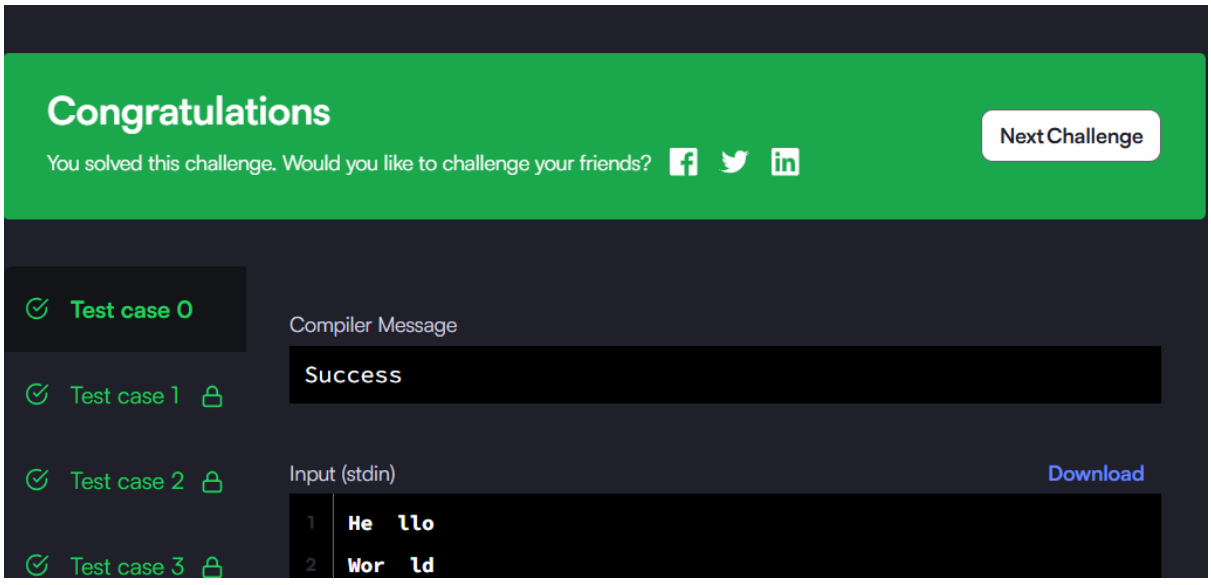

HackerRank
Prepare > Linux Shell > Text Processing > 'Tr' Command #2
Problem
Submissions
Leaderboard
In this challenge, we practice using the tr command because it is a useful translation tool in Linux.
In a given fragment of text, delete all the lowercase characters $a - z$.
Input Format
A block of ASCII text.
Output Format
Delete all the lowercase characters in the given block of text.
Sample Input
Hello
World
how are you
Sample Output
H
W

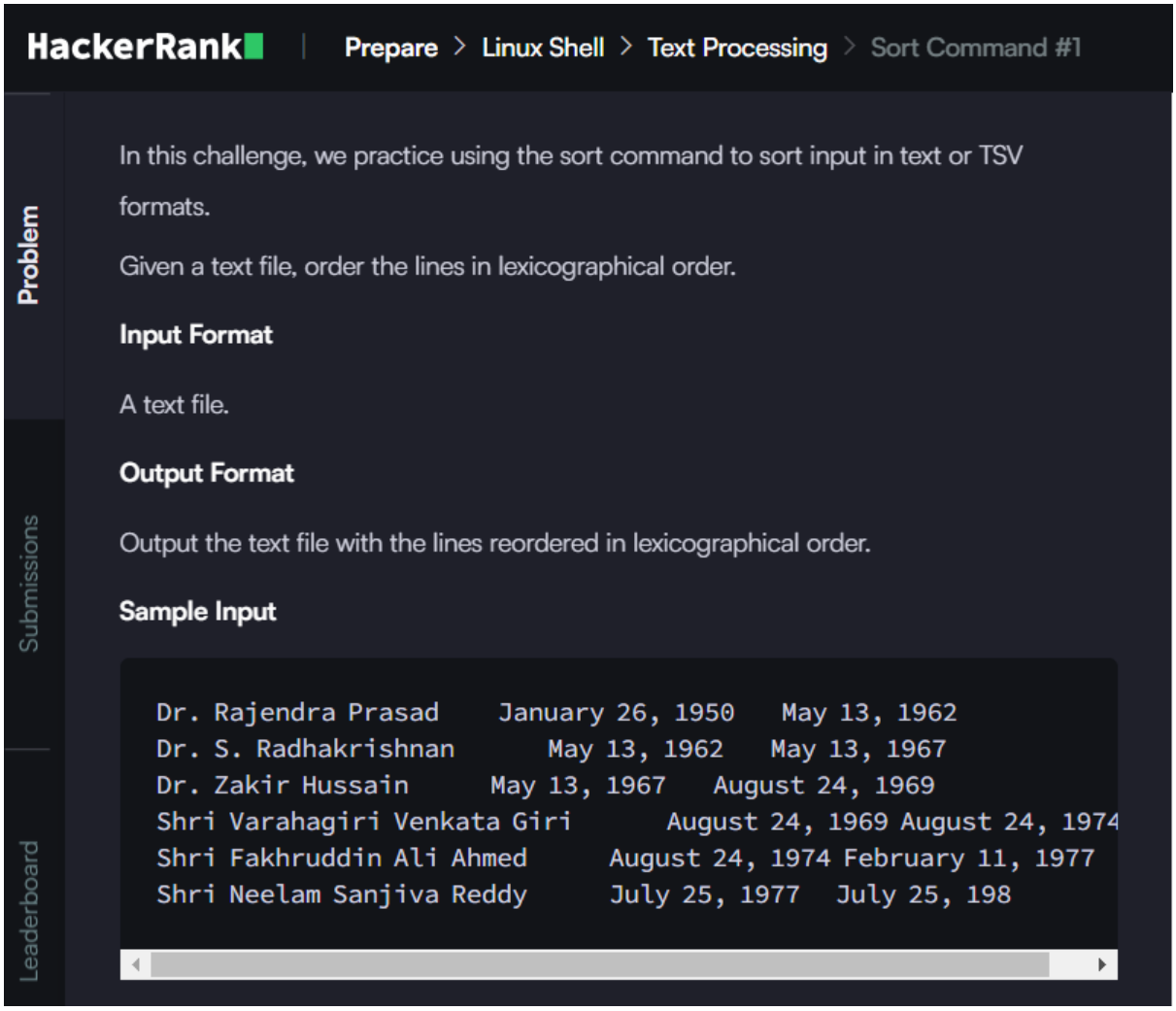

1 tr -d [a-z]


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Compiler Message
Success
Input (stdin)
Download
Hello
World

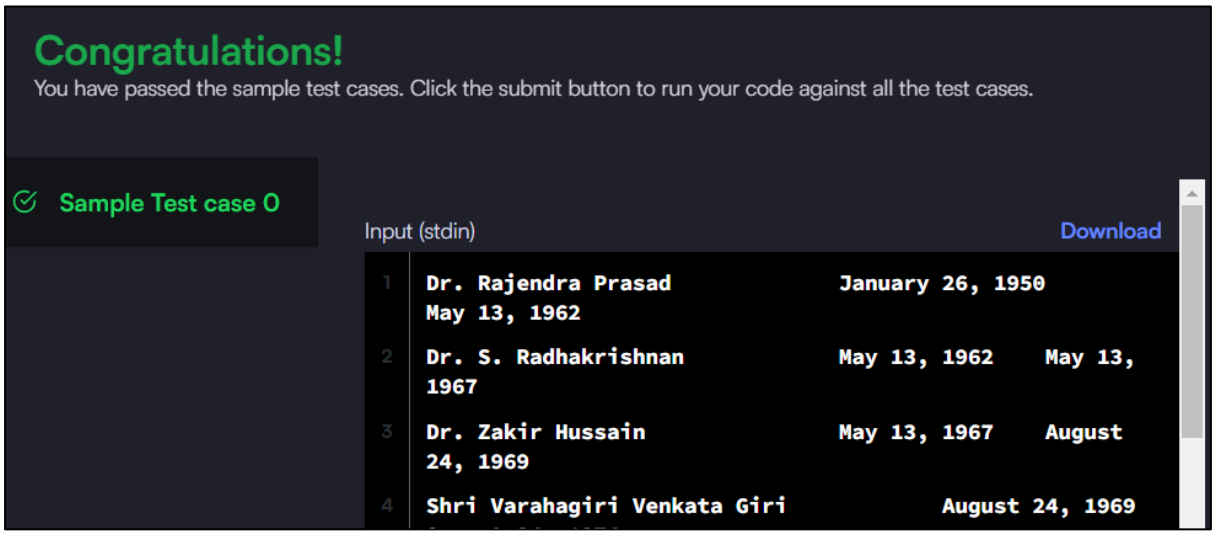

HackerRank
Prepare > Linux Shell > Text Processing > 'Tr' Command #3
Problem
Submissions
Leaderboard
In a given fragment of text, replace all sequences of multiple spaces with just one space.
Input Format
A block of ASCII text.
Output Format
Replace all sequences of multiple spaces with just one space.
Sample Input
He llo
Wor ld
how are you
Sample Output
He llo
Wor ld
how are you

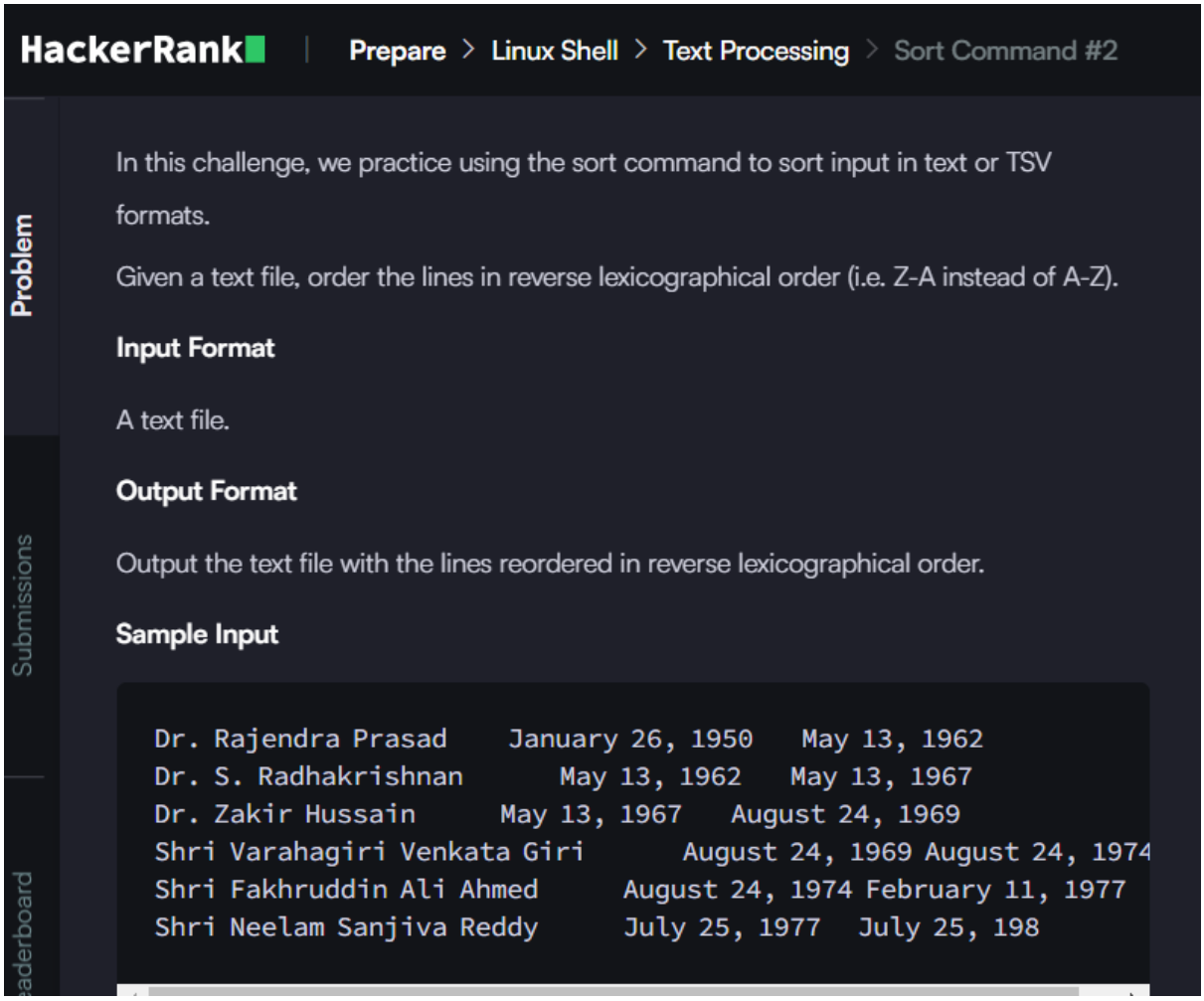

Change Theme
Languag
1
tr -s " "


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Compiler Message
Success
Input (stdin)
Download
He llo
Wor ld

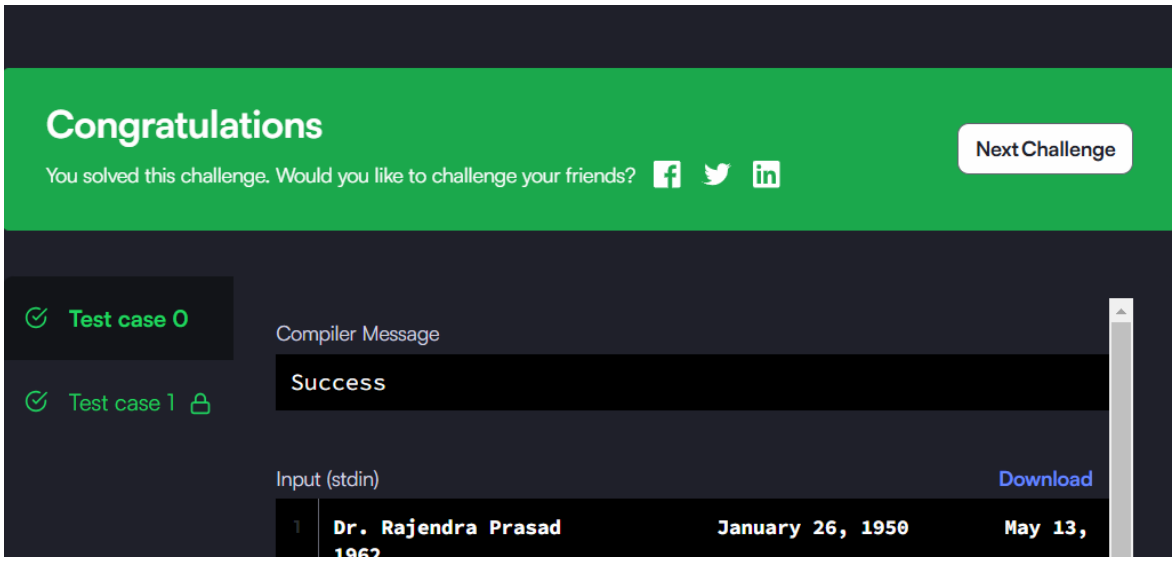

HackerRank
Prepare > Linux Shell > Text Processing > Sort Command #1
Problem
Submissions
Leaderboard
In this challenge, we practice using the sort command to sort input in text or TSV formats.
Given a text file, order the lines in lexicographical order.
Input Format
A text file.
Output Format
Output the text file with the lines reordered in lexicographical order.
Sample Input
Dr. Rajendra Prasad January 26, 1950 May 13, 1962
Dr. S. Radhakrishnan May 13, 1962 May 13, 1967
Dr. Zakir Hussain May 13, 1967 August 24, 1969
Shri Varahagiri Venkata Giri August 24, 1969 August 24, 1974
Shri Fakhruddin Ali Ahmed August 24, 1974 February 11, 1977
Shri Neelam Sanjiva Reddy July 25, 1977 July 25, 198

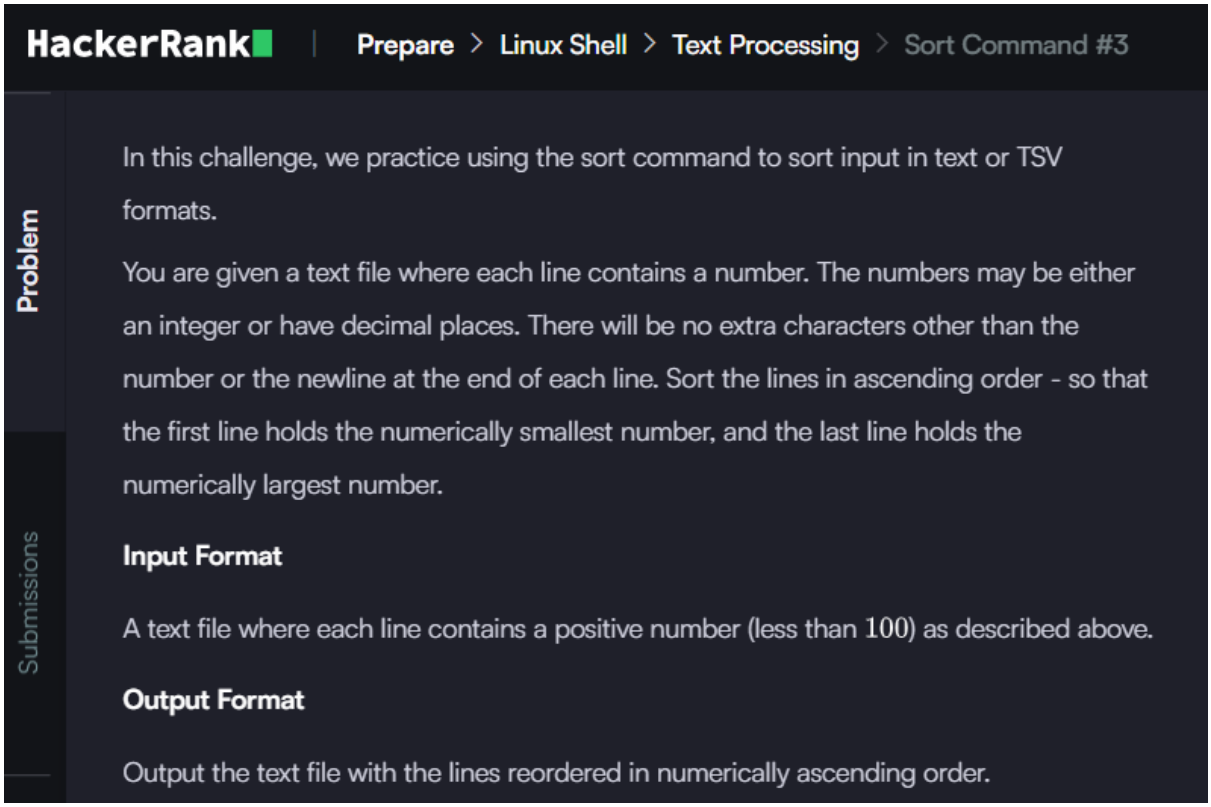

1
sort


Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
1 Dr. Rajendra Prasad January 26, 1950 May 13, 1962
2 Dr. S. Radhakrishnan May 13, 1962 May 13, 1967
3 Dr. Zakir Hussain May 13, 1967 August 24, 1969
4 Shri Varahagiri Venkata Giri August 24, 1969

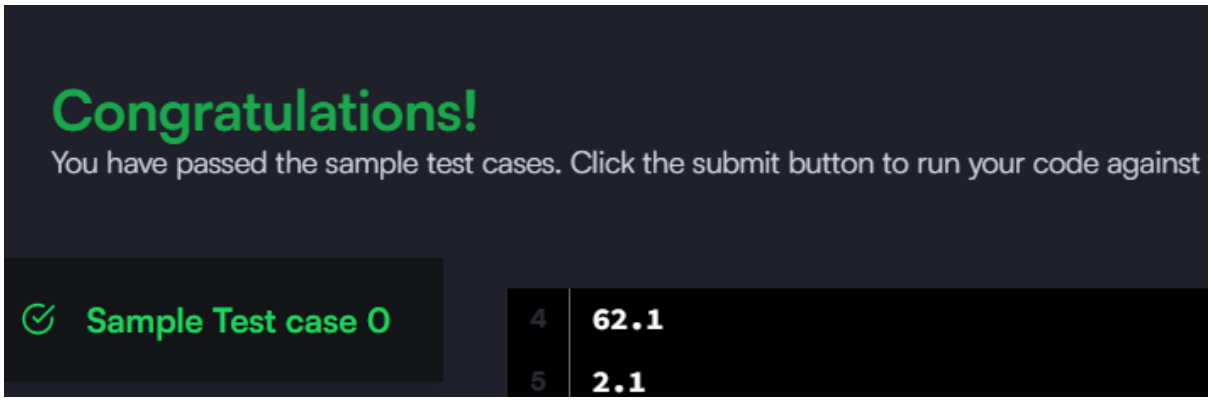

HackerRank
Prepare > Linux Shell > Text Processing > Sort Command #2
Problem
Submissions
Leaderboard
In this challenge, we practice using the sort command to sort input in text or TSV formats.
Given a text file, order the lines in reverse lexicographical order (i.e. Z-A instead of A-Z).
Input Format
A text file.
Output Format
Output the text file with the lines reordered in reverse lexicographical order.
Sample Input
Dr. Rajendra Prasad January 26, 1950 May 13, 1962
Dr. S. Radhakrishnan May 13, 1962 May 13, 1967
Dr. Zakir Hussain May 13, 1967 August 24, 1969
Shri Varahagiri Venkata Giri August 24, 1969 August 24, 1974
Shri Fakhruddin Ali Ahmed August 24, 1974 February 11, 1977
Shri Neelam Sanjiva Reddy July 25, 1977 July 25, 198

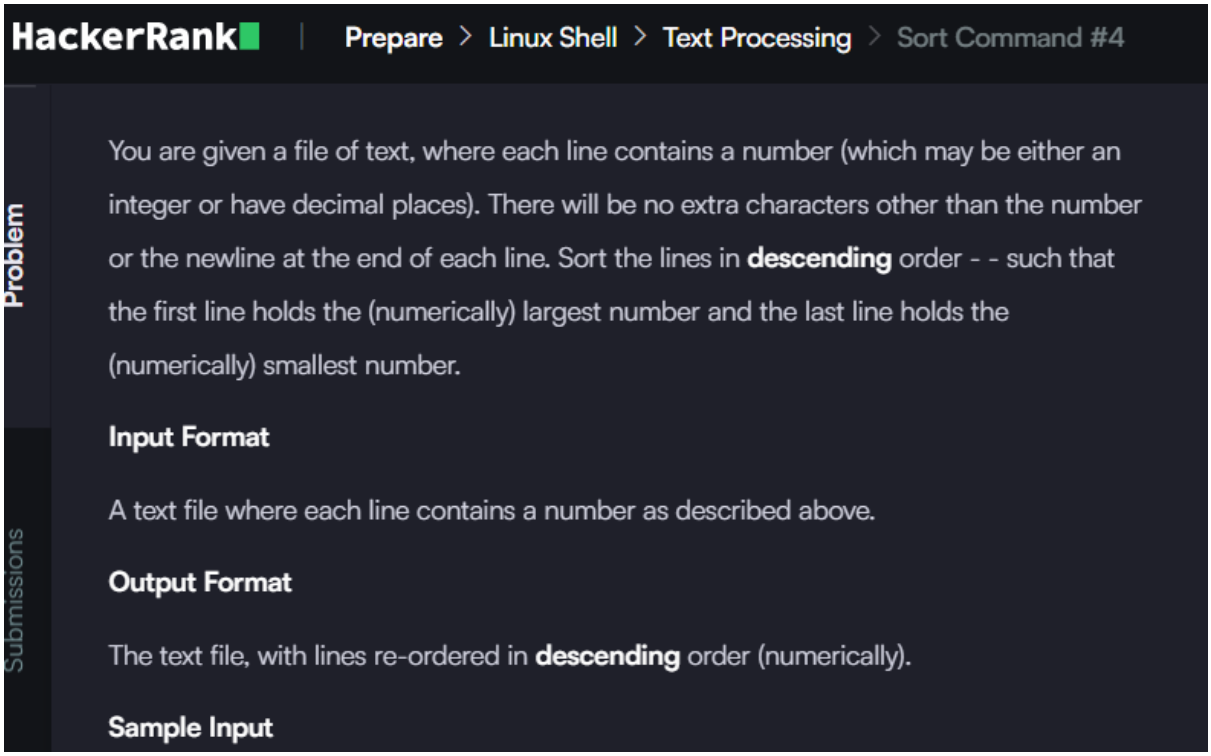

Change Theme
Languag
1
sort -r

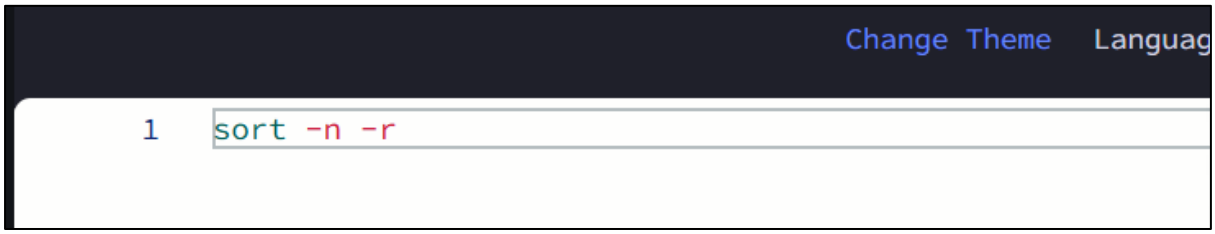

Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success
Input (stdin)
Download
1
Dr. Rajendra Prasad January 26, 1950 May 13,
1962

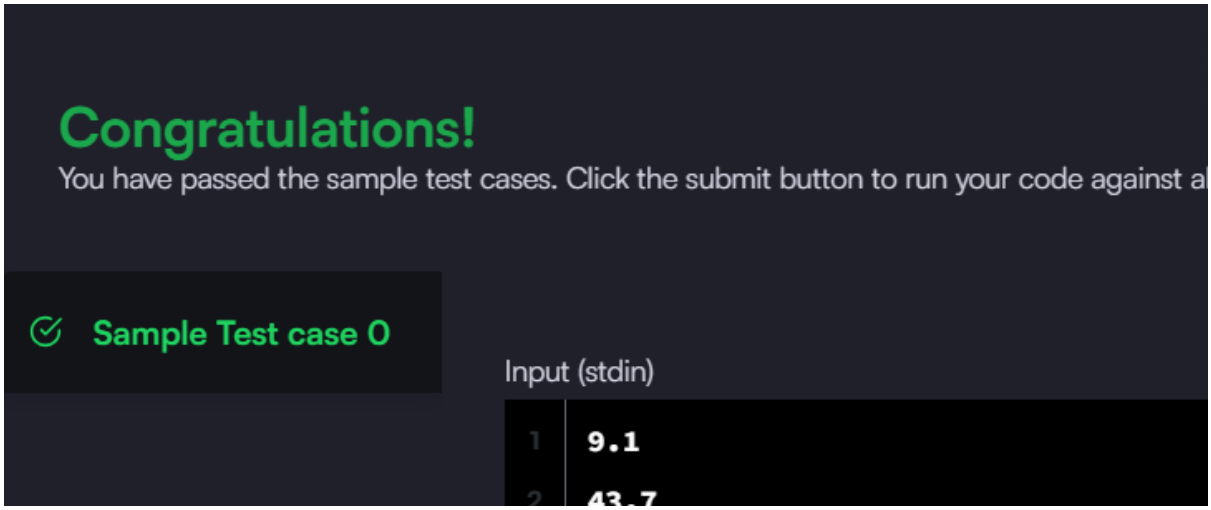

HackerRank
Prepare > Linux Shell > Text Processing > Sort Command #3
Problem
Submissions
In this challenge, we practice using the sort command to sort input in text or TSV formats.
You are given a text file where each line contains a number. The numbers may be either an integer or have decimal places. There will be no extra characters other than the number or the newline at the end of each line. Sort the lines in ascending order - so that the first line holds the numerically smallest number, and the last line holds the numerically largest number.
Input Format
A text file where each line contains a positive number (less than 100) as described above.
Output Format
Output the text file with the lines reordered in numerically ascending order.


Change Theme
Languag
1
sort -n


Congratulations!
You have passed the sample test cases. Click the submit button to run your code against
Sample Test case 0
4
62.1
5
2.1


HackerRank
Prepare > Linux Shell > Text Processing > Sort Command #4
Problem
Submissions
You are given a file of text, where each line contains a number (which may be either an integer or have decimal places). There will be no extra characters other than the number or the newline at the end of each line. Sort the lines in descending order - - such that the first line holds the (numerically) largest number and the last line holds the (numerically) smallest number.
Input Format
A text file where each line contains a number as described above.
Output Format
The text file, with lines re-ordered in descending order (numerically).
Sample Input


Change Theme
Languag
1
sort -n -r


Congratulations!
You have passed the sample test cases. Click the submit button to run your code against a
Sample Test case 0
Input (stdin)
1
9.1
2
43.7

HackerRank | Prepare > Linux Shell > Text Processing > Sort Command #5

Problem | Submissions | Leaderboard

You are given a file of text,which contains temperature information about American cities, in TSV (tab-separated) format. The first column is the name of the city and the next four columns are the average temperature in the months of Jan, Feb, March and April (see the sample input). Rearrange the rows of the table in **descending order** of the values for the average temperature in January.

Input Format

A text file where each line contains a row of data as described above.

Output Format

Rearrange the rows of the table in **descending order** of the values for the average temperature in January (i.e, the mean temperature value provided in the second column).

Sample Input 0

```
Albany, N.Y.  22.2  46.6  71.1  49.3  38.60  136  64.4  57
Albuquerque, N.M.  35.7  55.6  78.5  57.3  9.47  60  11.0  64
Anchorage, Alaska  15.8  36.3  58.4  34.1  16.08  115  70.8  39 / 60
Asheville, N.C.  35.8  54.1  73.0  55.2  47.07  126  15.3  39
Atlanta, Ga.  42.7  61.6  80.0  62.8  50.20  115  2.1  69 / 65
Atlantic City, N.J.  32.1  50.6  75.3  55.1  40.59  113  16.2  60 / 54
Austin, Texas  50.2  68.3  84.2  70.6  33.65  85  0.9  62 / 58
Baltimore, Md.  32.3  53.2  76.5  55.4  41.94  115  21.5  53
Baton Rouge, La.  50.1  66.6  81.7  68.1  63.08  110  0.2  52 / 46
```

```
1  sort -k2 -n -r -t$'\t'
2  # as here the file was TSV that is tab separated values, thats why we mentioned
   as a dellimeter so taht our command works and tab space is ignored
3
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
1  Albany, N.Y.  22.2  46.6  71.1  49.3  38.60
   136  64.4  57
2  Albuquerque, N.M.  35.7  55.6  78.5  57.3
   9.47  60  11.0  64
3  Anchorage, Alaska  15.8  36.3  58.4  34.1
   16.08  115  70.8  39 / 60
```

HackerRank | Prepare > Linux Shell > Text Processing > 'Sort' command #6

Problem

Submissions

board

You are given a file of tab separated weather data (TSV). There is no header column in this data file.

The first five columns of this data are: (a) the name of the city (b) the average monthly temperature in Jan (in Fahreneit). (c) the average monthly temperature in April (in Fahreneit). (d) the average monthly temperature in July (in Fahreneit). (e) the average monthly temperature in October (in Fahreneit).

You need to sort this file in ascending order of the second column (i.e. the average monthly temperature in January).

Input Format

A text file with multiple lines of tab separated data. The first five fields have been explained above

Output Format

Sort the data in ascending order of the average monthly temperature in January.

```
1  sort -k2 -n -t$'\t'
2  # as here the file was TSV that is tab separated values, thats why we mentioned
   as a dellimeter so that our command works and tab space is ignored
3
```

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

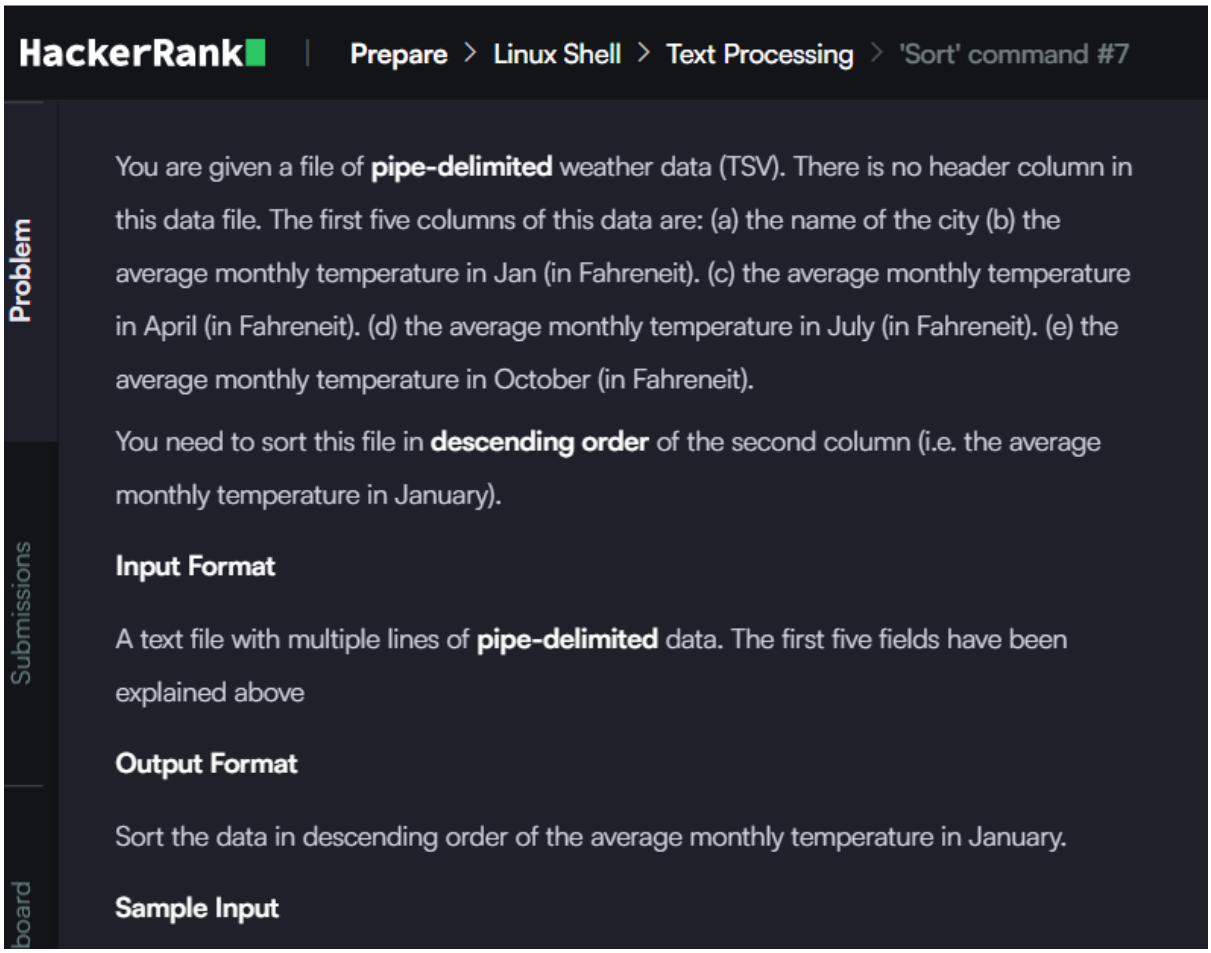
Test case 0

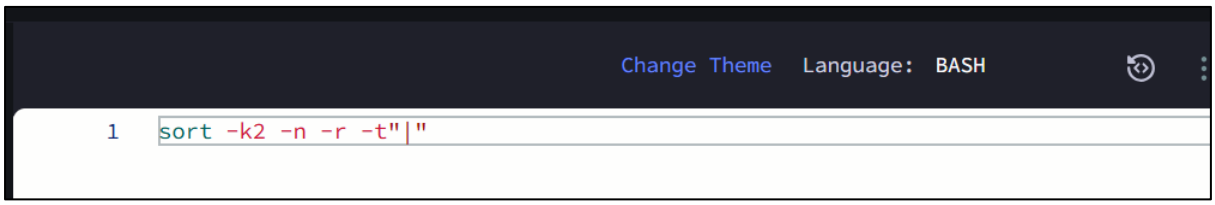
Test case 1

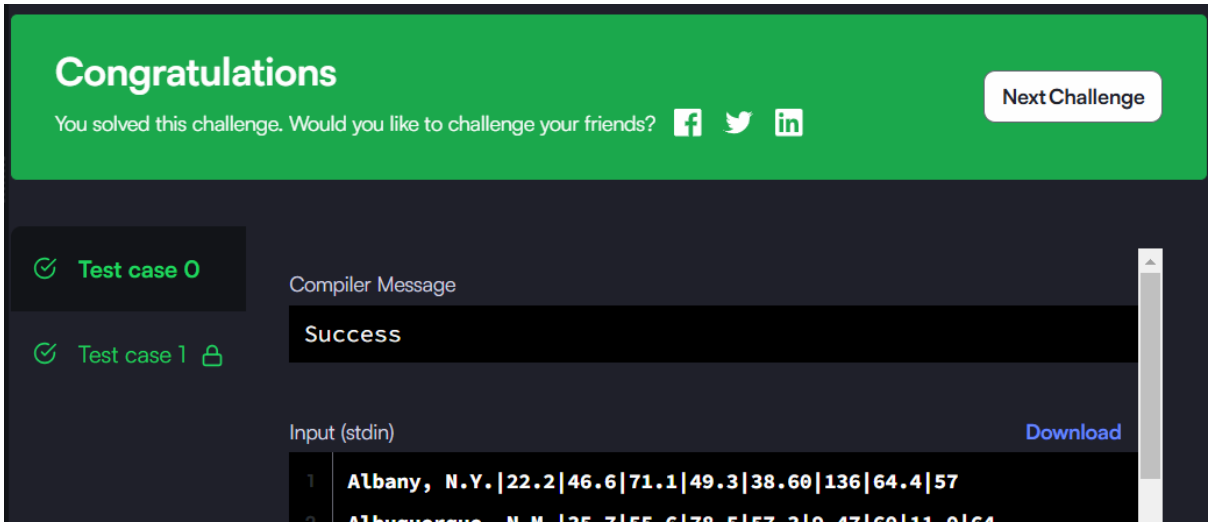
Success

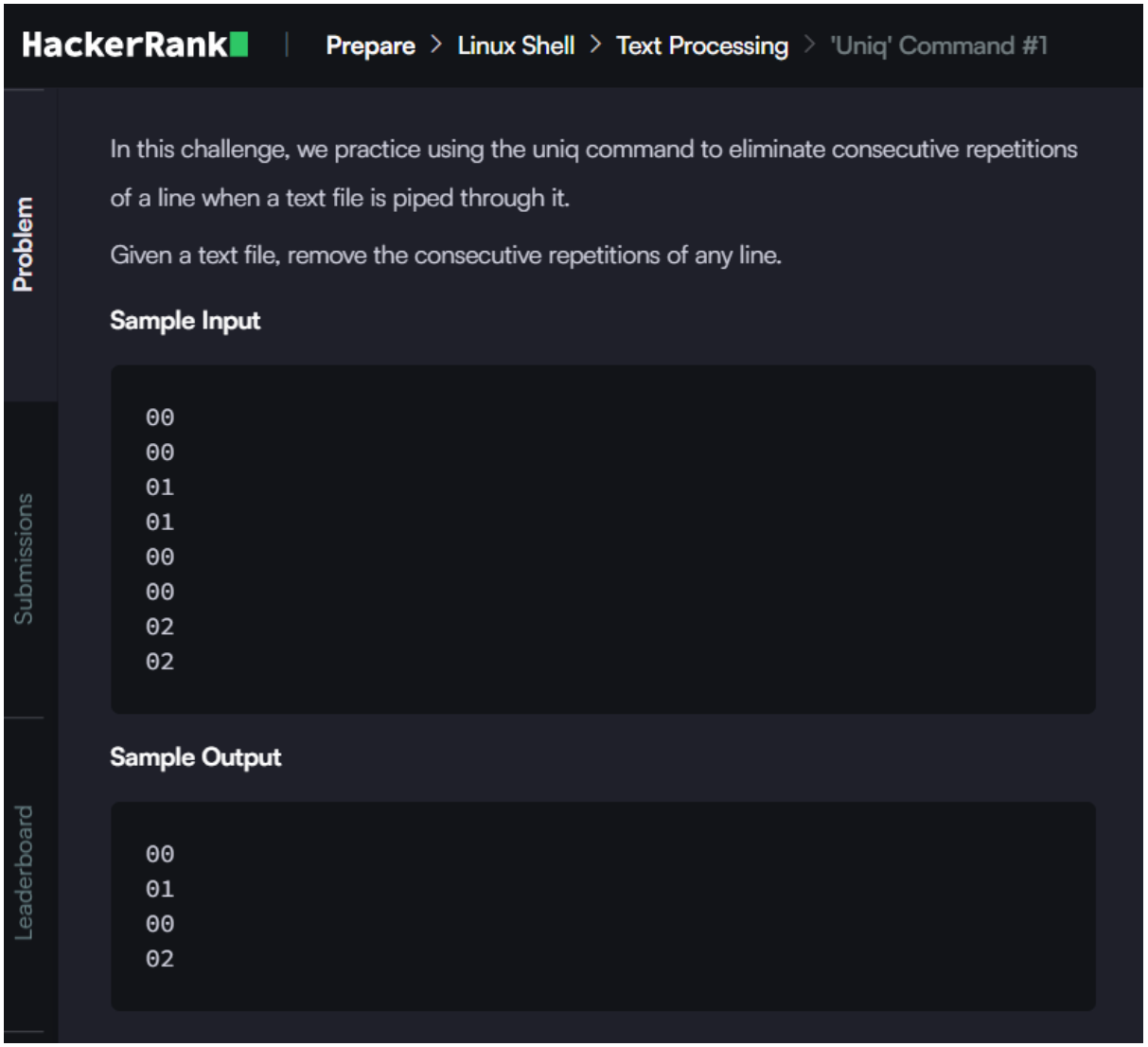
Input (stdin) Download

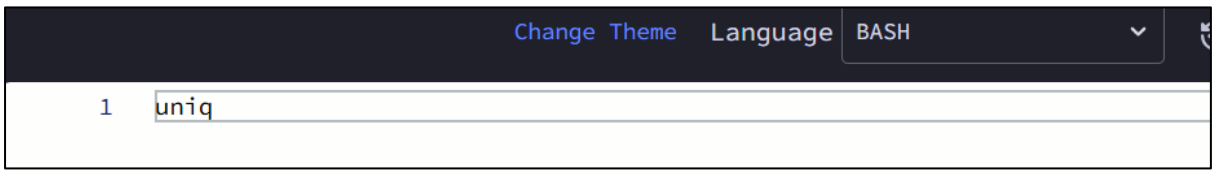
```
1  Albany, N.Y.    22.2    46.6    71.1    49.3    38.60    136
   64.4    57
2  Albuquerque, N.M.       35.7    55.6    78.5    57.3    9.47
   60      11.0    64
```

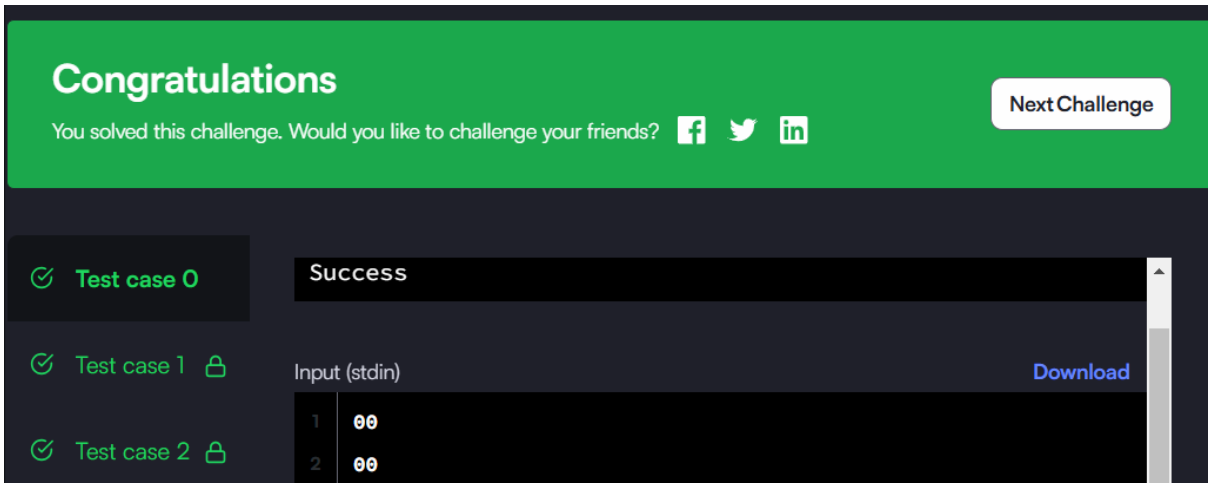

HackerRank
Prepare > Linux Shell > Text Processing > 'Sort' command #7
Problem
Submissions
board
You are given a file of pipe-delimited weather data (TSV). There is no header column in this data file. The first five columns of this data are: (a) the name of the city (b) the average monthly temperature in Jan (in Fahreneit). (c) the average monthly temperature in April (in Fahreneit). (d) the average monthly temperature in July (in Fahreneit). (e) the average monthly temperature in October (in Fahreneit).
You need to sort this file in descending order of the second column (i.e. the average monthly temperature in January).
Input Format
A text file with multiple lines of pipe-delimited data. The first five fields have been explained above
Output Format
Sort the data in descending order of the average monthly temperature in January.
Sample Input


Change Theme
Language: BASH
1
sort -k2 -n -r -t"|"


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success
Input (stdin)
Download
Albany, N.Y.|22.2|46.6|71.1|49.3|38.60|136|64.4|57


HackerRank
Prepare > Linux Shell > Text Processing > 'Uniq' Command #1
Problem
Submissions
Leaderboard
In this challenge, we practice using the uniq command to eliminate consecutive repetitions of a line when a text file is piped through it.
Given a text file, remove the consecutive repetitions of any line.
Sample Input
00
00
01
01
00
00
02
02
Sample Output
00
01
00
02


Change Theme
Language
BASH
1
uniq


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Success
Input (stdin)
Download
1
00
2
00

In this challenge, we practice using the uniq command to eliminate consecutive repetitions of a line when a text file is piped through it.

Given a text file, count the number of times each line repeats itself. Only consider consecutive repetitions. Display the space separated count and line, respectively. There shouldn't be any leading or trailing spaces. Please note that the uniq -c command by itself will generate the output in a different format than the one expected here.

Sample Output

Explanation

Sample Input

```
00
00
01
01
00
00
02
02
03
aa
aa
aa
```