

Bash Practice Problems

1 Query Processing

Write a bash script that first takes a query element and then an array of any length as command line argument. If the query element is present in the array, output "YES", else "NO".

Usage: `bash query_processing.sh {query_element} {array_of_elements}`

Sample Input 1: `bash query_processing.sh 2 3 7 8 2 6`

Output: YES

Explanation: 2 is present at the 4th position of the array {3, 7, 8, 2, 6}.

Sample Input 2: `bash query_processing.sh 10 1 2 4 6`

Output: NO

Explanation: 10 is not present in the array {1, 2, 4, 6}

Note: One solution would be to iterate over all the elements and then check the query presence. If you love challenges, sort the array using merge sort and use binary search to find the element. This will help decrease time complexity for large number of queries. (Don't worry, hopefully you will be able to do this after you have completed the assignment!)

2 Word Frequency Counter

Write a bash script that takes an input file, containing a list of words and outputs the frequency of each word. (Assume words to be separated by spaces.)

Usage: `bash word_frequency_counter.sh {input_file} > output.txt`

input_file:

apple banana orange mango banana apple apple mango

output.txt:

apple: 3

banana: 2

orange: 1

mango: 2

input_file:

Welcome to IITB!

Welcome to CS104!

output.txt:

Welcome: 2

to: 2

IITB!: 1

CS104!: 1

3 Bubble Sort

Write a bash script that takes an array of any length as an argument and outputs the sorted array. For sorting, use bubble sort algorithm. Your script should also print the number of swap operations performed. If you don't know about Bubble sort, please refer the [link](#).

Usage: bash bubble_sort.sh {array_of_elements}

Sample Input 1: bash bubble_sort.sh 1 4 3 2 6 4 7

Output:

1 2 3 4 4 6 7

4

Explanation: On the first line, the sorted array is printed. The 4 in the next line represent the number of swap operations, our code performed while sorting. (In this case, they are: (3, 4), (2, 4), (4, 6) and (2, 3).)

Sample Input 2: bash bubble_sort.sh 1 2 3 4 5 6 5

Output:

1 2 3 4 5 5 6

1

4 Recursive Functions

Write a bash script that takes an integer as command line argument and return the corresponding Catalan number. Catalan numbers are a sequence of natural numbers involved in various types of counting problems. One of the recursive forms of Catalan numbers is:

$$C_n = \sum_{i=0}^{n-1} C_i \times C_{n-i-1}$$

with $C_0 = 1$ and $C_1 = 1$. (There is another form of recursion for catalan numbers, find it out and write a bash script for that too!)

Usage: bash recursive_function.sh {integer}

Sample Input: bash recursive_function.sh 2

Output: 2

5 File Renaming

Write a bash script that creates one directory "output" and copies all the files {file_name}{date}.jpg present in the given directory. Now, here is how you should rename those files:

1. Assume that today's date is 26 November 2008.
2. The format for {date} is ddmmyyyy
3. {file_name} can be arbitrary large, but it will contain upper and lower case characters only.
4. Now you need to calculate the difference between today's date and the date of the file name.
5. If the date is 26112008, append "_today" at the end of file name.
6. If the date is between 19112008 and 25112008 (both inclusive), append "_weekold" at the end of the file name.
7. If the date is 18112008 or before, append "_quiteold" at the end of the file name.
8. If the date is 27112008 or later, delete all such files.

Usage: bash file_rename.sh

Testcase Explained: Let the current directory(say desktop) contain files:

- Attack26112008.jpg
- Time24112008.txt
- 30122007.jpg
- Weird19112008.jpg
- century15082047.jpg
- financialcrisis15092008.jpg

Now, the bash script creates a folder "output" inside desktop and that folder should contain the following files:

- Attack26112008_today.jpg
- 30122007_quiteold.jpg
- Weird19112008_weekold.jpg
- financialcrisis15092008_quiteold.jpg