

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Российский химико-технологический университет имени Д.И.  
Менделеева»

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Выполнил студент группы КС-36: Золотухин Андрей Александрович

Ссылка на репозиторий: [https://github.com/  
CorgiPuppy/  
info-sys-admin-labs](https://github.com/CorgiPuppy/info-sys-admin-labs)

Принял: Митричев Иван Игоревич

Дата сдачи: 19.03.2025

Москва  
2025

# Оглавление

Описание и выполнение задачи . . . . .	1
Задание А . . . . .	1
Задание Б . . . . .	1

# Описание и выполнение задачи

## Задание А

Определите размер самого большого файла и файла с самым длинным именем в директории */etc* (и поддиректориях), принадлежащих пользователю *root* (написать скрипт *bash*).

```
1 #!/bin/bash
2
3 declare TARGET_DIR=/etc
4 declare USER=root
5
6 set -v
7 find $TARGET_DIR -type f -user $USER 2> >(grep -v 'Permission denied$' 1>&2) -exec ls -
   1 {} + | head -n 1 <(sort -nr -k5) | awk '{print $9, $5}'
8
9 declare longest_file=$(find $TARGET_DIR -type f -user $USER 2> >(grep -v 'Permission
   denied$' 1>&2) -exec bash -c 'echo "$(basename "{}" | wc -c) {}"' \; | sort -nr |
   head -n 1 | cut -d' ' -f2-)
10 stat -c "%s %n" "$longest_file"
```

## Задание Б

Реализовать программу *bash* «pole\_chudes.sh», которая позволяет пользователю отгадывать слово по описанию, вводя отдельные буквы (слово берёт компьютер случайно из файла, подготовленного пользователем, где есть слов и их описания).

В случае ввода ранее введённой буквы программа выдаёт сообщение, что пользователь уже называл эту букву.

Программа должна показывать слово с отгаданными буквами.

```
1 #!/bin/bash
2
3 declare INPUT_FILE=src/task2/input.txt
4 declare -i var='cut -d ' ' -f 1 <(wc -l $INPUT_FILE)'
5 declare -i randomLine
6 declare -a answerReal
7 declare -a answerSuggested
8 usingLetters=""
9
10 if test -e $INPUT_FILE
11 then
12     varTemp="t"
13     varExit="q"
14
15     while [ "$varTemp" != "$varExit" ]
16     do
17         randomLine=$((($RANDOM % $var + 1))
18         answerRealString='cut -d '|' -f 1 <(tail -1 <(head -n $randomLine <(cat $INPUT_FILE
   )))'
19         task='cut -d '|' -f 2 <(tail -1 <(head -n $randomLine <(cat $INPUT_FILE)))'
20         echo "Task: $task."
21
22         for (( i=0; i<${#answerRealString}; i++ ))
```

```

23 do
24     answerReal[$i]=${answerRealString:i:1}
25     answerSuggested[$i]="_"
26 done
27
28 echo "1. Amount of characters: ${answerSuggested[@]}."
29
30 while [[ "${answerSuggested[@]}" != "${answerReal[@]}" ]]
31 do
32     read -p "2. Suggest your letter: " suggestedLetter
33     if [[ ${#suggestedLetter} -eq 1 ]]
34     then
35         if [[ $usingLetters != *$suggestedLetter* ]]
36         then
37             usingLetters="${usingLetters}${suggestedLetter:0:1}"
38
39             if [[ ${answerReal[@]} =~ $suggestedLetter ]]
40             then
41                 echo "'$suggestedLetter' was found."
42                 for index in `seq ${#answerReal[@]}`
43                 do
44                     char="${answerRealString:(( index - 1 )):1}"
45                     [[ "$char" != "$suggestedLetter" ]] && continue || answerSuggested[index
- 1]=$char
46                 done
47             else
48                 echo "'$suggestedLetter' wasn't not found."
49             fi
50
51             echo ${answerSuggested[@]}
52         else
53             echo "You have already called '$suggestedLetter'!"
54         fi
55     else
56         echo "You've called too much letters!"
57     fi
58 done
59
60 while [[ -n ${answerSuggested[@]} ]]
61 do
62     answerSuggested=("${answerSuggested[@]:1}")
63 done
64 while [[ -n ${answerReal[@]} ]]
65 do
66     answerReal=("${answerReal[@]:1}")
67 done
68 usingLetters=""
69
70 echo "3. If you want to exit you can enter 'q':"
71 read varTemp
72 done
73 else
74     echo "$INPUT_FILE doesn't exist!"
75 fi

```