

ОТЧЕТ (МЕЛЬНИКОВ НИКИТА ИУ10-36) BASH ЗАДАЧИ

#1

Написать скрипт, отображающий всю информацию о файле из текущего рабочего каталога (файл и каталог придумать самим)

```
#!/bin/bash
ls -l zad1file.txt
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad1.sh
-rw-rw-r-- 1 nikita nikita 47 Nov 29 16:04 zad1file.txt
nikita@DESKTOP-BNU31HG:~$
```

#2

Шаблон

```
1
1 2
1 2 3
1 2 3 4
```

Считайте значение от пользователя и выведите вышеуказанную лесенку

```
#!/bin/bash

read -p "Введите число строк лесенки (ladder): " n

for ((i = 1; i <= n; i++)); do
    for ((j = 1; j <= i; j++)); do
        echo -n "$j "
    done
    echo
done
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad2.sh
Введите число строк лесенки (ladder): 6
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

#3

Шаблон

```
1
2 3
4 5 6
7 8 9 10
```

Считайте значение от пользователя и выведите вышеуказанную лесенку

```
#!/bin/bash

read -p "Введите число строк для лестницы из чиселок: " n

number=1

for ((i = 1; i <= n; i++)); do
    for ((j = 1; j <= i; j++)); do
        echo -n "$number "
        ((number++))
    done
    echo
done
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad3.sh
Введите число строк для лестницы из чиселок: 6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

#4

В этом задании вам нужно просто определить, является ли данный год високосным или нет. Если вы не знаете правил, вот они:

- Годы, кратные 4, являются високосными,
- но годы, кратные 100, не являются високосными,
- но годы, делящиеся на 400, являются високосными.

Годы находятся в пределах допустимого $1600 \leq \text{year} \leq 4000$.

```
#!/bin/bash

read -p "Введите год для определения на високосность: " year

if ((year >= 1600 && year <= 4000)); then
    if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then
        echo "$year - високосный год"
    else
        echo "$year - невисокосный год"
    fi
else
    echo "Введенный год должен быть в пределе от 1600 до 4000."
fi
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad4.sh
Введите год для определения на високосность: 1800
1800 - невисокосный год
nikita@DESKTOP-BNU31HG:~$ ./zad4.sh
Введите год для определения на високосность: 2024
2024 - високосный год
nikita@DESKTOP-BNU31HG:~$ ./zad4.sh
Введите год для определения на високосность: 1500
Введенный год должен быть в пределе от 1600 до 4000.
```

5

Печать шахматной доски заданной при помощи передаваемого параметра

Чтобы распечатать черный квадрат, echo -e -n "\\e[40m" " "

Чтобы напечатать белое квадрат, echo -e -n "\\e[47m" " "

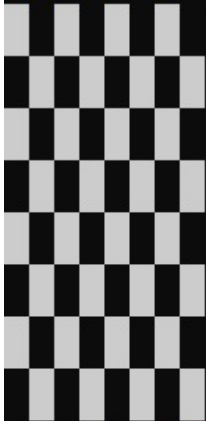
Вызов команд происходит в цикле.

```
#!/bin/bash

read -p "Введите размер шахматной доски в виде числа: " n

for ((i = 1; i <= n; i++)); do
    for ((j = 1; j <= n; j++)); do
        if (( (i + j) % 2 == 0 )); then
            echo -e -n "\e[47m "
        else
            echo -e -n "\e[40m "
        fi
    done
    echo -e -n "\e[0m "
    echo
done
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad5.sh
Введите размер шахматной доски в виде числа: 8
```



6

Описание:

Ваша задача состоит в том, чтобы сложить буквы в одну букву.

Функции будет предоставлено переменное количество аргументов, каждый из которых представляет собой букву для добавления.

Notes:

Буквы всегда будут строчными.

Буквы могут переполняться (см. предпоследний пример описания)

Если буквы не указаны, функция должна вернуть 'z'

Примеры:

```
addLetters('a', 'b', 'c') = 'f'
addLetters('a', 'b') = 'c'
addLetters('z') = 'z'
addLetters('z', 'a') = 'a'
addLetters('y', 'c', 'b') = 'd' // notice the letters overflowing
addLetters() = 'z'
```

```
#!/bin/bash

addLetters() {
    if [ "$#" -eq 0 ]; then
        echo 'z'
        return
    fi

    sum=0

    for letter in "$@"; do
        ascii=$(printf "%d" "'$letter")
        sum=$((sum + ascii - 96))
    done

    sum=$(( (sum - 1) % 26 + 1))
    result=$(printf \\$(printf '%03o' $((sum + 96))))
    echo "$result"
}

addLetters "$@"
```

```
nikita@DESKTOP-BNU31HG:~$ ./zad6.sh a b c
f
nikita@DESKTOP-BNU31HG:~$ ./zad6.sh a b
c
nikita@DESKTOP-BNU31HG:~$ ./zad6.sh a z
a
```

7

Напишите алгоритм, который будет определять действительные адреса IPv4 в десятичном формате с точками.
IP-адреса следует считать действительными, если они состоят из четырех октетов со значениями от 0 до 255 включительно.

Примеры допустимых входных данных:

```
1.2.3.4
123.45.67.89
```

Недопустимые примеры ввода:

```
1.2.3
1.2.3.4.5
123.456.78.90
123.045.067.089
```

Заметки:

Начальные нули (например, 01.02.03.04) считаются недействительными.
Входные данные гарантированно будут одной строкой

```

is_valid_() {
    local ip="$1"
    local
    reg="^([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$"

    if [[ $ip =~ $reg ]]; then
        echo "Действительный адрес"
    else
        echo "Недействительный адрес"
    fi
}

ip_="$1"
is_valid_ "$ip_"

```

```

nikita@DESKTOP-BNU31HG:~$ ./zad7.sh 127.5.5.2
Действительный адрес
nikita@DESKTOP-BNU31HG:~$ ./zad7.sh 127.5.05.2
Недействительный адрес
nikita@DESKTOP-BNU31HG:~$ ./zad7.sh 127.5.5.300
Недействительный адрес

```