


<p><b>POLITECHNIKA WROCŁAWSKA</b></p>  <p>Wydział Informatyki i Telekomunikacji</p>	<p>Wydział: Informatyki i Telekomunikacji</p> <p>Kierunek: Cyberbezpieczeństwo</p> <p>Rok Akademicki: 2024/2025</p> <p>Rok studiów, semestr: I, 2</p> <p>Grupa: 2</p> <p>Termin: <i>poniedziałek, godz. 15.15</i></p>
<p><b>Programowanie skryptowe – Laboratorium 3</b></p>	
<p>Prowadzący:</p> <p>mgr inż. Karolina Pfajfer</p> <p>Data wykonania ćwiczenia:</p> <p>17.03.2025</p> <p>Data oddania sprawozdania:</p> <p>23.03.2025</p>	<p>Autor:</p> <p><i>Adam Dąbrowski, 283832</i></p>

## 1. Zmienne

a) Napisz skrypt wymagający podania imienia i wieku w argumencie, przypisz do zmiennych, a następnie wyświetl te zmienne w formie: „Imie: , Wiek: ”

```
GNU nano 8.2          osoby.sh *
```

```
#!/bin/bash
read -p "Podaj imie " miejsce
read -p "Podaj rok urodzenia " rok

obecny_rok=$(date +%Y)
wiek=$((obecny_rok-rok))
echo "Imie: $miejsce"
echo "Wiek: $wiek"
```

```
(userkali@hostkali)-[~]
└─$ ./osoby.sh
Podaj imie Adam
Podaj rok urodzenia 2004
Imie: Adam
Wiek: 21
```

b) Utwórz skrypt, który pobiera od użytkownika długość i szerokość prostokąta. Oblicza pole powierzchni prostokąta i wyświetli wynik na ekranie.

```
GNU nano 8.2                                prostokat.sh *
read -p "podaj jeden i drugi bok prostokata" bok1 bok2

pole=$((bok1*bok2))
echo "pole to $pole"
```

```
(kali㉿kali)-[~]
$ ./prostokat.sh
podaj jeden i drugi bok prostokata 4 6
pole to 24
```

## 2. Instrukcje warunkowe

a) Napisz skrypt który zapyta o nazwę użytkownika i sprawdzi czy dany użytkownik to admin. Jeżeli tak zwróć odpowiedź „Konto admina”, jeżeli nie zwróć „To nie jest konto admina”

#W trakcie tworzenia sprawozdania musiałem przeinstalować maszynę dlatego kali@kali zmienia się na userkali@hostkali

```
GNU nano 8.2                                admin.sh
#!/bin/bash

read -p "podaj konto do sprawdzenia" nazwa

if [[ $nazwa = $HOSTNAME ]]
then
echo "Konto admina"
else
echo "To nie jest konto admina"
fi
```

```
(userkali㉿hostkali)-[~]
$ ./admin.sh
podaj konto do sprawdzenia adam
To nie jest konto admina

(userkali㉿hostkali)-[~]
$ ./admin.sh
podaj konto do sprawdzenia hostkali
Konto admina
```

b) Napisz skrypt, który zapyta o nazwę pliku, a następnie sprawdzi czy plik istnieje, czy jest pusty, czy jest katalogiem i wypisze odpowiedź na każde pytanie w osobnej linii.

```
(userkali@hostkali)-[~]  
$ ./info_o_kat.sh  
jaki plik sprawdzic pusty.txt  
plik istanieje  
plik jest pusty  
plik nie jest katalogiem
```

```
(userkali@hostkali)-[~]  
$ ./info_o_kat.sh  
jaki plik sprawdzic pelny.txt  
plik istanieje  
plik nie jest pusty  
plik nie jest katalogiem
```

```
(userkali@hostkali)-[~]  
$ ./info_o_kat.sh  
jaki plik sprawdzic kat_pusty  
plik istanieje  
plik nie jest pusty  
plik jest katalogiem
```

```
(userkali@hostkali)-[~]  
$ ./info_o_kat.sh  
jaki plik sprawdzic fol  
plik istanieje  
plik jest pusty  
plik jest katalogiem
```

```
(userkali@hostkali)-[~]  
$ ./info_o_kat.sh  
jaki plik sprawdzic pusty  
Plik nie istnieje  
Plik jest pusty  
Plik nie jest katalogiem
```

```

read -p "jaki plik sprawdzic " plik

if [[ -n $plik ]]
then
    lokalizacja=$(locate $plik | awk 'NR==1')
    #echo $lokalizacja
    if [[ $plik = $(echo $lokalizacja | rev | cut -d '/' -f 1 | rev) ]]
    then
        rozmiar=$(ls -l $lokalizacja | awk '{print $5}')
        #echo $rozmiar
        if [[ -d $lokalizacja ]]
        then
            #echo "katalog"
            #czy_kat_pusty=$(ls -A $lokalizacja)
            if [[ $(ls -A $lokalizacja) ]]
            then
                echo "plik istanieje"
                echo "plik nie jest pusty"
                echo "plik jest katalogiem"
            else
                echo "plik istanieje"
                echo "plik jest pusty"
                echo "plik jest katalogiem"
            fi
        else
            if [[ $rozmiar -eq 0 ]]
            then
                echo "plik istanieje"
                echo "plik jest pusty"
                echo "plik nie jest katalogiem"
            else
                echo "plik istanieje"
                echo "plik nie jest pusty"
                echo "plik nie jest katalogiem"
            fi
        fi
    else
        echo "Plik nie istnieje"
        echo "Plik jest pusty"
        echo "Plik nie jest katalogiem"
    fi
fi

```

### 3. Operatory logiczne

a) Napisz skrypt, który zapyta o liczbę, a następnie przy pomocy operatora AND sprawdzi czy liczba jest dodatnia i czy jest nieparzysta, a na koniec wyświetli wynik

```

(userkali@hostkali)-[~]
$ ./dod_parz.sh
Podaj liczbe7
liczba jest dodatnia i nieparzysta

(userkali@hostkali)-[~]
$ ./dod_parz.sh
Podaj liczbe6
liczba nie jest dodatnia lub nieparzysta

(userkali@hostkali)-[~]
$ ./dod_parz.sh
Podaj liczbe0
liczba nie jest dodatnia lub nieparzysta

(userkali@hostkali)-[~]
$ ./dod_parz.sh
Podaj liczbe-1
liczba nie jest dodatnia lub nieparzysta

```

```

GNU nano 8.2 dod_parz.sh
#!/bin/bash
read -p "Podaj liczbe" liczba

if [ $((($liczba%2)) -ne 0) ] && [ "$liczba" -gt 0 ]
then
echo "liczba jest dodatnia i nieparzysta"
else
echo "liczba nie jest dodatnia lub nieparzysta"
fi

```

b) Napisz skrypt, który poprosi użytkownika o podanie nazwy użytkownika i hasła, a następnie sprawdzi, czy podane dane są poprawne (np. nazwa użytkownika to "admin" i hasło to "password").

```

GNU nano 8.2 uwierzytelnianie.sh
#!/bin/bash

read -p "podaj login " login
read -p "podaj haslo " haslo

if [ $login = "admin" ] && [ $haslo = "password" ]
then
echo "dane sa poprawne"
else
echo " dane nie sa poprawne"
fi

```

```

(userkali@hostkali)-[~]
$ ./uwierzytelnianie.sh
podaj login admin
podaj haslo password
dane sa poprawne

(userkali@hostkali)-[~]
$ ./uwierzytelnianie.sh
podaj login adam
podaj haslo haslo
dane nie sa poprawne

```

#### 4. Pętle

a) Napisz skrypt generujący 10 adresów IP za pomocą pętli for, które będą różniły się czwartym oktetem rosnąco w adresie 192.168.1.X

```

(userkali@hostkali)-[~]
$ ./ros_adresyIP.sh
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
192.168.0.6
192.168.0.7
192.168.0.8
192.168.0.9
192.168.0.10

```

```

GNU nano 8.2
#!/bin/bash

for i in {1..10}
do
echo "192.168.0.$i"
done

```

b) Napisz ten sam skrypt za pomocą pętli while

```

(userkali@hostkali)-[~]
$ ./ros_adresyIP.sh
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
192.168.0.6
192.168.0.7
192.168.0.8
192.168.0.9
192.168.0.10

```

```
GNU nano 8.2
#!/bin/bash

#for i in {1..10}
#do
#echo "192.168.0.$i"
#done
i=1
while [[ $i -le 10 ]]
do
    echo "192.168.0.$i"
    ((i++))
done
```

## 5. Funkcje

a) Utwórz skrypt wyświetlający informacje o systemie: Aktualna data, Wersja linuxa, Użytkownik, Adres IP. Każda informacja powinna znajdować się w osobnej funkcji. Użyj jednej zmiennej z nazwą komputera w każdej funkcji. Przykład: „Aktualna data na (Nazwa komputera) to X”

```
(userkali@hostkali)~[~]
$ ./aktualnerzeczy.sh
Aktualna data hostkali Thu Mar 20 03:57:40 PM EDT 2025
Obecna wersja hostkali Linux hostkali 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kal
i 6.11.2-1kali1 (2024-10-15) x86_64 GNU/Linux
Zalogowany uzytkownik na hostkali to userkali
Adres IP na hostkali to 10.0.2.15
```

```
GNU nano 8.2          aktualnerzeczy.sh
#!/bin/bash

data() {
echo "Aktualna data $HOSTNAME $(date)"
}
linux() {
echo "Obecna wersja $HOSTNAME $(uname -a)"
}
uzytkownik(){
echo "Zalogowany uzytkownik na $HOSTNAME to $(whoami)"
}
IP(){
echo "Adres IP na $HOSTNAME to $(hostname -I | awk '{print $1}')"
}

data
linux
uzytkownik
IP
```



Zadanie dodatkowe lab3:

```
GNU nano 8.2
#!/bin/bash
#wszystkie zmienne
obecna_data=$(date +"%Y.%m.%d %H:%M")
ram=$(free -h | awk 'NR==2 {print $2}')
dysk=$(df -h --output=source,used,avail)
host=$HOSTNAME
system=$(uname -o)
File System
█
#raport koncowy

cat << TEKST > system_info.txt
Data raportu
$obecna_data

_____

Całkowita ilość pamięci RAM w systemie:
$ram

_____

Wykorzystanie dysków:
$dysk

_____

Nazwa hosta:
$host

_____

Nazwa systemu operacyjnego:
$system

TEKST
```



Trash  
(userkali@hostkali)-[~]  
\$ ./dodatkowe\_lab3.sh

(userkali@hostkali)-[~]  
\$ cat system\_info.txt

Data raportu  
2025.03.23 09:07

---

Całkowita ilość pamięci RAM w systemie:  
7.8Gi

Home

---

Wykorzystanie dysków:

Filesystem	Used	Avail
udev	0	3.9G
tmpfs	976K	794M
/dev/sda1	14G	8.2G
tmpfs	4.0K	3.9G
tmpfs	0	5.0M
tmpfs	0	1.0M
tmpfs	0	1.0M
tmpfs	0	1.0M
tmpfs	0	1.0M
tmpfs	0	1.0M
tmpfs	288K	3.9G
tmpfs	0	1.0M
tmpfs	0	1.0M
tmpfs	112K	795M

---

Nazwa hosta:  
hostkali

---

Nazwa systemu operacyjnego:  
GNU/Linux