

1. Zmienne

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

```
#Z1a
if [ $# != 2 ]; then
    echo "Podaj argumenty!"
fi
imie=$1
wiek=$2
echo "Imie: $imie, Wiek: $wiek"
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh Patryk 21
Imie: Patryk, Wiek: 21
```

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

```
#Z1b
read -p "Podaj dlugosc boku prostokata: " dlugosc
read -p "Podaj szerokosc prostokata: " szerokosc
echo "Pole powierzchni wynosi: $((dlugosc * szerokosc))"
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj dlugosc boku prostokata: 4
Podaj szerokosc prostokata: 5
Pole powierzchni wynosi: 20
```

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”

```
# Z2a
read -p "Podaj nazwe uzytkownika: " usern
if [ -z $usern ]; then
    echo "Nie podano nazwy!"
    exit
fi
if [[ -n $(groups $usern | grep sudo) || "$usern" == "root" ]]; then
    echo "Konto admina"
else
    echo "To nie jest konto admina"
fi
```

```
(kali@kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj nazwe uzytkownika: kali
Konto admina
```

- c) 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.

```
#Z2b
read -p "Podaj sciezke do pliku: " sciezka

if [ -e $sciezka ]; then
    echo "Plik istnieje"
    #
    if [ -s $sciezka ]; then
        echo "Plik nie jest pusty"
    else
        echo "Plik jest pusty"
    fi
    #
    if [ -d $sciezka ]; then
        echo "Plik jest katalogiem!"
    else
        echo "Plik nie jest katalogiem!"
    fi
    #
else
    echo "Plik nie istnieje"
fi
```

```
(kali@kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj sciezke do pliku: /home/kali/arp_detection.py
Plik istnieje
Plik nie jest pusty
Plik nie jest katalogiem!
```

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

```
#Z3a
read -p "Podaj liczbe: " liczba

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

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj liczbe: 11
Liczba jest dodatnia i nieparzysta
```

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").

```
# #Z3b

read -p "Podaj nazwe uzytkownika: " username
read -p "Podaj haslo: " pass

if [[ "$username" == "admin" && "$pass" == "password" ]]; then
    echo "Podano prawidlowe dane!"
else
    echo "Podano nieprawidlowe dane!"
fi
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj nazwe uzytkownika: admin
Podaj haslo: password
Podano prawidlowe dane!
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
Podaj nazwe uzytkownika: ktos
Podaj haslo: cos
Podano nieprawidlowe dane!
```

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

```
# #Z4a

tablica=()

for ((i=1; i<11; i++)); do
    tablica[$i-1]="192.168.1.$i"
done
echo "${tablica[@]}"
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
192.168.1.1 192.168.1.2 192.168.1.3 192.168.1.4 192.168.1.5 192.168.1.6 192.168.1.7
192.168.1.8 192.168.1.9 192.168.1.10
```

b) Napisz ten sam skrypt za pomocą pętli while

```
#Z4b
tablica=()
counter=1
while [ $counter -lt 11 ]; do
    tablica[$counter-1]="192.168.1.$counter"
    ((counter++))
done
echo "${tablica[@]}"
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
$ ./skrypty.sh
192.168.1.1 192.168.1.2 192.168.1.3 192.168.1.4 192.168.1.5 192.168.1.6 192.168.1.7
192.168.1.8 192.168.1.9 192.168.1.10
```

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”

```
# #Z5

NAME=$(hostname)

function data() {
    echo "Aktualna data na $NAME to: $(date)"
}
function wersja() {
    echo "Aktualna wersja na $NAME to: $(uname -r)"
}
function uzytkownik() {
    echo "Aktualny uzytkownik na $NAME to: $(whoami)"
}
function adresIP() {
    echo "Aktualny adres IP na $NAME to: $(ip route get 1.1.1.1 | awk '{print $(NF-2); exit}' )"
}
data
wersja
uzytkownik
adresIP
```

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe]
└─$ ./skrypty.sh
Aktualna data na kali to: Thu Mar 20 08:22:52 AM EDT 2025
Aktualna wersja na kali to: 6.8.11-amd64
Aktualny uzytkownik na kali to: kali
Aktualny adres IP na kali to: 192.168.131.128
```