

# System operacyjny Linux

## Lista nr 7

1. Napisz w bash-u skrypt, która pyta użytkownika o imię, nazwisko, adres i telefon, a następnie wypisuje wprowadzone dane w „ładnym” formacie. Przykładowe działanie programu:

```
# ./z1.sh
Podaj imię: John
Podaj nazwisko: Kowalski
Podaj adres: ul. Cicha 15, 55-555 Wrocław
Podaj telefon: 607 123 456
John Kowalski, zamieszkały ul. Cicha 15, 55-555 Wrocław, tel. 607 123 456
```

2. Napisz w bash-u skrypt, który prosi użytkownika po podawanie kolejnych liczb. Wczytywanie kończy w momencie, gdy użytkownik poda liczbę zero i wtedy wypisuje sumę podanych wcześniej liczb. Zakładamy, że użytkownik jest grzeczny i podaje liczby (a nie np. imię misia Yogi). Przykładowe działanie programu:

```
# ./z2.sh
Podaj liczbę: 2
Podaj liczbę: 3
Podaj liczbę: 4
Podaj liczbę: 0
Suma: 9
```

3. Napisz w bash-u skrypt, które będzie działał analogicznie do polecenia `seq` umówionego na wykładzie (z opcji wystarczy zaimplementować `-s`).
4. Napisz w bash-u skrypt, który będzie robił backup pewnego katalogu. Skrypt powinien działać następująco:
  - Katalog do zarchiwizowania powinien być przekazywany jako parametr wywołania programu.
  - Nazwa pliku z archiwum powinna być następująca: `backup-rok-miesiac-dzien-godzina-minuta.tar.bz2` (rozszerzenia implikują jakich narzędzi należy użyć).
  - Skrypt ma sprawdzać, czy plik o nazwie, która ma być nazwą archiwum już istnieje. Jeśli istnieje, ma skończyć działanie ze stosownym komunikatem. Ponadto skrypt ma także sprawdzać, czy istnieje katalog, który ma zostać zarchiwizowany. Jeśli nie istnieje, również ma skończyć działanie ze stosownym komunikatem.
5. Napisz w bash-u skrypt, który utworzy konta na podstawie pliku o formacie

```
konto1
haslo1
opis1
konto2
haslo2
opis2
...
```

Utworzone konta powinny być ważne przez rok.

*Wskazówka: w rozwiązaniu skorzystać z pętli `for` i polecenia `chpasswd`. Ułatwieniem może być utworzenie dodatkowego pomocniczego pliku dla polecenia `chpasswd`.*

- 6.\* Napisz w bash-u skrypt, który bierze parametry będące liczbami, a w wyniku drukuje określone przez te parametry liczby Fibbonaciego<sup>1</sup>. W przypadku, gdyby parametr był większy od 40, jako wynik podaje liczbę -1.<sup>2</sup>

```
# ./z4 10 20 30 40 50
89
10946
1346269
165580141
-1
```

*Paweł Rajba*

---

<sup>1</sup>Liczby Fibonacciego definiujemy następująco:  $F(0) = 0$ ;  $F(1) = 1$ ;  $F(n) = F(n - 1) + F(n - 2)$ , dla  $n \geq 2$

<sup>2</sup>Zadanie dodatkowe dla tych, którzy zrobili wszystkie pozostałe zadania.