

## Lekcja 4 – Praca domowa

**1. (2 pkt.)** Napisać skrypt, który wszystkie pliki z przyrostkiem ~ (np. plik.txt~), skopiuje (jeżeli takie są) do katalogu BACKUP w bieżącym katalogu. Jeżeli katalog BACKUP nie istnieje, skrypt powinien go założyć. Jeżeli jest już plik (lub inny nie-katalog) o nazwie BACKUP, skrypt powinien zgłosić błąd.

```
#!/bin/bash
#Kopiuje pliki z przyrostkiem .txt do katalogu BACKUP

#Sprawdza czy istnieje nie-katalog BACKUP
if [[ -f BACKUP ]] ; then
echo Błąd: plik BACKUP już istnieje, kopiowanie nieudane.
exit 1
fi

#Sprawdza czy istnieje katalog BACKUP, jeśli nie to zakłada
if [[ ! -d /BACKUP ]] ; then
echo Katalog BACKUP nie istnieje, zakładam nowy.
mkdir BACKUP
fi

#wypisujemy liste plikow plik*.txt w katalogu
for PLIK in *.txt ; do

#Kopiuje kazdy plik do katalogu
cp $PLIK BACKUP
done

#Wyswietl komunikat o udanym kopiowaniu
echo Kopiowanie udane.

exit 0
```

**2. (2 pkt.)** Napisać skrypt, który obliczy n-tą liczbę Fibonacciego - rekurencyjnie.

**#Wyciągnij n-tą liczbę Fibonacciego**

```
fibor(){
  if [ $1 -le 0 ]; then
    echo 0
  elif [ $1 -eq 1 ]; then
    echo 1
  else
    echo `fibor $[$1-2]` + `fibor $[$1 - 1]`
  fi
}
```

**#Wywołujemy funkcję z argumentem \$1**  
fibor \$1

**3. (2 pkt.) Napisać skrypt, który obliczy n-tą liczbę Fibonacciego - iteracyjnie.**

```
#!/bin/bash
#Wylicza n-ta liczbe Fibonacciego - iteracyjnie

#Inicjujemy pierwsza liczbe ciagu
LICZBA_A=0

#Inicjujemy druga liczbe ciagu
LICZBA_B=1

#Inicjujemy nastepna liczbe ciagu (początkowo trzecia)
LICZBA=$((LICZBA_A + LICZBA_B))

#Drukujemy kolejne liczby ciagu
for (( i=3; i <= $1; i++ )); do
    LICZBA_A=LICZBA_B
    LICZBA_B=LICZBA
    LICZBA=$((LICZBA_A + LICZBA_B))
done
echo $LICZBA
exit 0

#Wywołujemy skrypt z argumentem $1
```

**4. (2 pkt.) Napisać skrypt, który wczyta plik zawierający liczby (po jednej w wierszu) i wypisze ich maksimum, minimum i sumę. Jeśli plik jest pusty, to wypisze odpowiedni komunikat. Nazwa pliku ma być podana jako parametr skryptu.**

```
#!/bin/bash
#Wczytuje plik z liczbami podany jako argument i wypisuje max, min i sume

#liczniki: max, min i suma
MAX=0
MIN=999
SUMA=0

#Sprawdzamy czy plik jest pusty
if [ ! -s $1 ] ; then
    echo Plik jest pusty
    exit 1
fi

#Dla kazdej linii w pliku
for LICZBA in `cat $1` ; do

#Aktualizuj sume
SUMA=$((SUMA + LICZBA))
```

```
#porownaj wielkosc liczby z maximum
if [ $MAX -lt $LICZBA ] ; then
MAX=$LICZBA
fi

#porownaj wielkosc liczby z minimum
if [ $MIN -gt $LICZBA ] ; then
MIN=$LICZBA
fi
done

echo Najwieksza liczba w $1 wynosi $MAX
echo Najmniejsza liczba w $1 wynosi $MIN
echo Suma liczb w $1 wynosi $SUMA
exit 0
```

**5. (2 pkt.) Napisać skrypt rozwiązujący zadanie 4 dla wielu plików podanych jako argumenty. Wykorzystaj podprogramy.**

```
#!/bin/bash
#Wczytuje pliki z liczbami podanymi jako argumenty skryptu i wypisuje max, min i sume

#Definiujemy funkcje oblicz()
oblicz () {
#resetujemy liczniki: max, min i suma
MAX=0
MIN=999
SUMA=0

#Sprawdzamy czy plik jest pusty
if [ ! -s $ARG ] ; then
echo Plik $ARG jest pusty
exit 1
else
#Dla kazdej linii w pliku
for LICZBA in `cat $ARG` ; do

#Aktualizuj sume
SUMA=$((SUMA + $LICZBA))

#porownaj wielkosc liczby z maximum
if [ $MAX -lt $LICZBA ] ; then
MAX=$LICZBA
fi

#porownaj wielkosc liczby z minimum
if [ $MIN -gt $LICZBA ] ; then
```

```
MIN=$LICZBA
```

```
fi
```

```
done
```

```
fi
```

```
}
```

```
#Definiujemy funkcje drukuj()
```

```
drukuj () {
```

```
#Drukujemy wynik
```

```
echo Najwieksza liczba w $ARG wynosi $MAX
```

```
echo Najmniejsza liczba w $ARG wynosi $MIN
```

```
echo Suma liczb w $ARG wynosi $SUMA
```

```
}
```

```
#Glowna petla skryptu, iterujemy po argumentach skryptu
```

```
for ARG in $@ ; do
```

```
oblicz $ARG
```

```
drukuj $ARG
```

```
done
```

```
exit 0
```