Wczytanie danych z klawiatury do zmiennej:

```
read zmienna
```

aby potem np. echo $zmienna.
Ze względu na fakt, że wiele skryptów nie jest interaktywnych i służą automatyzacji zadań, jest to wykorzystywane w sytuacjach awaryjnych (np. szybkie zapytanie użytkownika o kontynuację).

date
Instrukcja drukuje na terminal bieżący czas i datę. Można użyć ją w np. echo:

```
echo dzisiaj jest `date`
```

Uruchamianie programu w tle:

```
program &
```

np.

```
less tekst.txt &
```

Program zostanie automatycznie odesłany do tła. Na konsolę zostanie wydrukowany [numer zadania] i PID. **Po wylogowaniu** i tak programy te zostaną „ubite”.

Case

```
case zmienna in
wartosc1)
. . .
;;
wartosc2)
. . .
;;
*)
. . .
;;
esac
```

Blok instrukcji **case** kończymy słowem **esac**. Np.

```
#!/bin/bash
 . . .
case $1 in
set)
export a=2
./skrypt2.sh
;;
unset)
export a=0
./skrypt2.sh
;;
*)
echo "Niewlasciwe polecenie!"
;;
esac
```

Polecenie export ustawia nam tutaj zmienną tak, by inne skrypty ją widziały. Widzi ją wywołany skrypt2.sh.

Instrukcja **test** lub **[**
Stosowana w if czy while.

```
test wyrażenie
      lub:
[ wyrażenie ]
```

Zwraca 0 (prawda) lub wynik różny od 0 (fałsz).

! - negacja
-a – AND
-o – OR

If – sprawdzanie warunku:

```
if test….
then
 ….
fi
```

albo, krócej:

```
if [ … ]
then
….
fi
```

Tak jak w przypadku for, można stosować if […]; then. Zestaw instrukcji pod if'em zamykamy słowem **fi**.

Możliwe warunki:

Liczbowe:

[a -eq b]	- a == b
[a -lt b]	- a < b
-gt	- a > b
-le	- a <= b
-ge	- a >= b
-ne	- a != b

Łańcuch tekstu:

[-z $str]	- prawda, gdy ciąg pusty
[-n $str]	- prawda gdy dłuższy od 0.
[$str1 = $str2]	- porównania
!=	- prawda gdy nierówny

Pliki

[-s plik.txt]	- plik istnieje i ma rozmiar >0
[-f plik]	- plik istnieje i jest plikiem
[-d plik]	- plik istnieje, ale jest katalogiem
[-r plik]	- plik istnieje i mamy prawo do jego odczytu

podobnie -w, -x
[plik1 -nt plik2] - plik1 jest nowszy niż plik2
-ot – starszy niż

Zadania:

Napisz skrypt działający z argumentem 1: start/stop. Użyj instrukcji case.

- w przypadku start: uruchom program wczytany jako argument 2,w pliku log zapisz komunikat:
  program nnn uruchomiono: date
- w przypadku stop: zatrzymaj w/w program przez killall i zapisz w logu:
  Program zatrzymano: date.

Uwzględnij sytuację, w której użytkownik poda nieprawidłowe argumenty. Położenie pliku log przechowaj w zmiennej wewnątrz skryptu.