2.6 Edytor sed (Stream Editor)

sed to edytor do manipulowania plikami tekstowymi. Jako edytor „strumieniowy” różni się od innych, takich jak **vi** lub **gedit**.

sed nie pracuje interaktywnie, ale jest kontrolowany opcjami polecenia liniowego lub skryptem takim, jak tryb **ex** w **vi**.

Edytor **sed** wczytuje bieżący wiersz pliku do wewnętrznego bufora celem manipulowania tekstem. Wynik jest wysyłany na standardowe wyjście. Oryginalny plik nie jest nigdy zmieniany. Jeżeli chcemy zapisać wyjście w pliku, należy przekierować standardowe wyjście na plik.

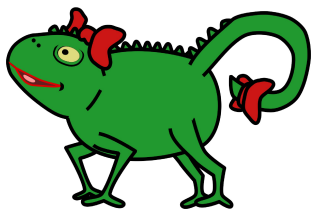
Składnia uruchamiania sed: sed *opcje „polecenie” nazwa_pliku*

- Najprostszym poleceniem **sed** jest „**p” (print):**

```
geeko@da51:~ > sed 'p' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
LOREM IPSUM DOLOR SIT AMET



Lorem ipsum dolor sit amet,
Lorem ipsum dolor sit amet,
...
```

Jak widać w powyższym przykładzie, każdy wiersz jest dwukrotnie drukowany, ponieważ **sed** domyślnie wysyła każdy wiersz na standardowe wyjście. By temu zapobiec, powinno się dodać opcję **-n**:

```
geeko@da51:~ > sed -n 'p' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET

Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
...
```



Przyjrzyjmy się bliżej części „*polecenie*” składni wywołania **sed**.

Przede wszystkim, można podać deklarację adresu z numerami wierszy, które **sed** powinien sprawdzić. Są różne możliwości:

brak deklaracji - wszystkie wiersze

numer - tylko wiersz o numerze *numer,* np:

```
geeko@da51:~ > sed -n '1p' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
```

start, end - wiersze od *start* do *end*, np:

```
geeko@da51:~ > sed -n '2,4p' loremipsum.txt

Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
```

$ - ostatni wiersz, np:

```
geeko@da51:~ > sed -n '36,$p' loremipsum.txt
facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit
praesent luptatum zzril delenit
au gue duis dolore te feugat nulla facilisi.
```

/regexp/ - wiersze zgodne z regularnym wyrażeniem *regexp*, np:

```
geeko@da51:~ > sed -n '/^D/p' loremipsum.txt
Duis te feugifacilisi.
Duis autem dolor in hendrerit
Duis te feugifacilisi per suscipit
Duis te feugifacilisi.
Duis autem dolor in hendrerit in
```

Wyjaśnienie. Zostaną wydrukowane wiersze zaczynające się od dużej litery „D”.

numer, /regexp/ - wiersze od *numer* do pierwszego wiersza, który spełnia kryterium *regexp*, np:

```
geeko@da51:~ > sed -n '9,/^D/p' loremipsum.txt
suscipit lobortis nisl
ut aliquip ex ea commodo consequat.
Duis te feugifacilisi.
```

Wyjaśnienie. Zostaną wydrukowane wszystkie wiersze od wiersza 9 do pierwszego wiersza zaczynającego się od dużej litery „D”.

Można też, w podobny sposób, użyć wyrażenia regularnego jako punktu początkowego dla **sed**.

Regularne wyrażenia edytora **sed** są podobne do tych omówionych wcześniej. Jednak są tu dodatkowe znaki specjalne wymagające poprzedzenia znakiem „ \”:

- **\(** … **|)**
- **\{** … **|}**

Można połączyć dwa lub więcej poleceń **sed** używając opcji **-e**:

```
geeko@da51:~ > sed -n -e '3p' -e '6p' -e '9p' loremipsum.txt
Lorem ipsum dolor sit amet,
ut lacreet dolore magna aliguam erat volutpat.
suscipit lobortis nisl
```

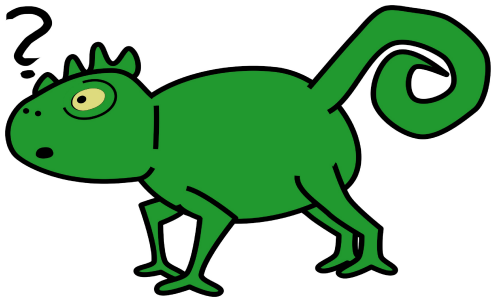
Wyjaśnienie. Wiersze 3, 6 oraz 9 zostaną wyświetlone na ekranie.

Można też rozdzielić polecenia używając znaku **„;”** :

```
geeko@da51:~ > sed -n '3p;6p;9p' loremipsum.txt
Lorem ipsum dolor sit amet,
ut lacreet dolore magna aliguam erat volutpat.
suscipit lobortis nisl
```

Ćwiczenie. Używanie sed – część I

Użyj sed do wyświetlenia na ekranie wierszy 10 do 15 z dowolnego pliku tekstowego.



Poniżej podano kilka innych ważnych poleceń **sed**:

- **„a” (append**) dołącza jeden lub więcej wierszy za podanym wierszem (wierszami) w poleceniu. Przykład:

```
geeko@da51:~ > sed '1aHello Geeko' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
Hello Geeko
Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
...
```

  Jeżeli chcesz dodać więcej niż jeden wiersz, należy znak nowego wiersza zamaskować znakiem ukośnika „\”:

```
geeko@da51:~ > sed '1aHello Geeko\
> Hello Suzie' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
Hello Geeko
Hello Suzie
Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
...
```

- **„i” (insert)** dodaje jeden lub więcej wierszy przed podanym (podanymi):

```
geeko@da51:~ > sed '1iHello Geeko' loremipsum.txt
Hello Geeko
LOREM IPSUM DOLOR SIT AMET
Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
...
```

- **„r nazwa_pliku”** wstawia zawartość podanego pliku za podanym wierszem (wierszami):

```
geeko@da51:~ > sed '1r numbers.txt' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
0000
123
...
98798
765765
Lorem ipsum dolor sit amet,
```

```
consectetuer adipiscing elit,
sed diem nonummy nibh euismodtincidunt
...
```

„d” (delete) kasuje wskazany wiersz (wiersze).

```
geeko@da51:~ > sed '1,3d' loremipsum.txt
consectetuer adipiscing elit,
sed diem nonummy nibh euismodtincidunt
ut lacreet dolore magna aliguam erat volutpat.
```

„c” (replace) zastępuje podanym tekstem wskazany wiersz (wiersze):

```
geeko@da51:~ > sed '1,3cHello Geeko' loremipsum.txt
Hello Geeko
consectetuer adipiscing elit,
sed diem nonummy nibh euismodtincidunt
ut lacreet dolore magna aliguam erat volutpat.
```

Wyjaśnienie. Wiersze 1 do 3 zostały zastąpione tekstem „Hello Geeko”.

„w” (write) zapisuje wyjście edytora **sed** do podanego pliku. Można użyć tego polecenia zamiast przekierowania wyjścia przez użycie **„>**”.

```
geeko@da51:~ > sed -n '1w output.txt' loremipsum.txt
```

Wyjaśnienie. Tworzony jest nowy plik o nazwie output.txt. Zawiera pierwszy wiersz pliku loremipsum.txt.

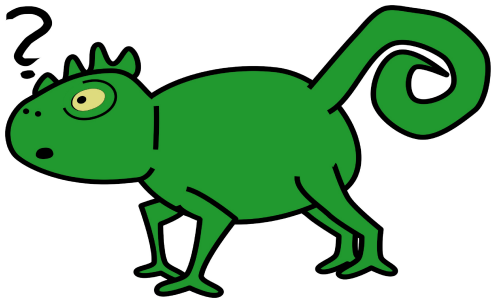
Jeżeli pominiemy opcję „**-n**”, zawartość loremipsum.txt zostanie wyświetlona na ekranie, a nie zapisana do pliku output.txt.

„q” (quit) kończy pracę z edytorem.

```
geeko@da51:~ > sed '3q' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET


Lorem ipsum dolor sit amet,
```

Ćwiczenie. Używanie edytora sed – część II

Użyj sed do usunięcia pierwszych dziesięciu linii wybranego pliku tekstowego oraz zapisania ich w pliku short.txt.



y/ *znak (i) /znak (i) /* *z*amienia pojedyncze znaki. Pierwszy znak z pierwszego zestawu zostaje zastąpiony pierwszym znakiem z drugiego zestawu, drugi znak z pierwszego zestawu – drugim znakiem z drugiego zestawu, i tak dalej – do wyczerpania list.

```
geeko@da51:~ > sed 'y/oim/XYZ/' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET

LXreZ YpsuZ dXlXr sYt aZet,
cXnsectetuer adYpYscYng elYt,
sed dYeZ nXnuZZy nYbh euYsZXdtYncYdunt
```

Wyjaśnienie. Małe „o” zostaje zastąpione przez „X”, „i” przez „Y”, a „m” przez „Z”.

Zamiast używania ukośnika, można użyć dowolnego innego znaku. Ostatnie polecenie może wyglądać następująco:

```
sed 'y?oim?XYZ?' loremipsum.txt
```

lub

```
sed 'yBoimBXYZB' loremipsum.txt
```

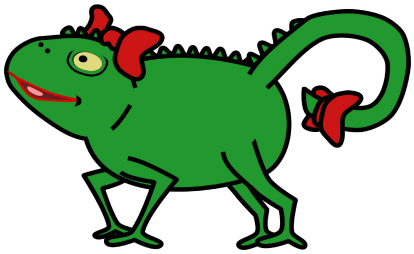
Znak po poleceniu „y” definuje którego znaku użyto jako separatora obu zestawów znaków i do zakończenia polecenia.

s *wzorzec /string/flag (s)* (substitute) zastępuje *wzorzec wyszukiwania* drugim wzorcem, określonym przez *string*. Zwykle, *wzorzec wyszukiwania* zawiera regularne wyrażenia.

```
geeko@da51:~ > sed 's/dol/XXXXX/' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET

Lorem ipsum XXXXXor sit amet,
consectetuer adipiscing elit,
```

```
sed diem nonummy nibh euismodtincidunt
ut lacreet XXXXXore magna aliguam erat volutpat.
...
```

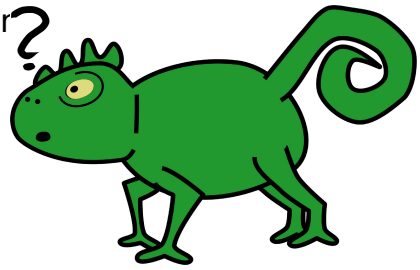
Wyjaśnienie. Kombinacja znaków „dol” zostanie zastąpiona przez „XXXXX”.



Ćwiczenie. Używanie edytora sed – część III

Użyj sed do zastąpienia „quis” przez „quod” w pliku loremipsum.txt.

Wyjście powinno być zapisane w pliku quodlorem.



Można użyć dowolnego znaku zamiast ukośnika.

Wyrażenia regularne są „zachłanne” i próbują objąć ciąg tak długi, jak tylko jest to możliwe.

Na przykład:

```
geeko@da51:~ > echo AABBCCAABB | sed s/A.*A/XX/
XXBB
geeko@da51:~ >
```

Wyjaśnienie. Regularne wyrażenie określa ciąg zaczynający się od „A” z następującymi po nim 0 lub więcej znaków i kończy się na „A”. Ponieważ wyrażenie regularne jest „zachłanne”, graniczne „A” to nie są „A” sąsiednie (pierwsze i drugie) tylko obejmujące jak największy ciąg znaków.

Domyślnie „s” zastępuje tylko pierwszy znaleziony, zgodny ze wzorcem wyszukiwania, ciąg. Można kontrolować zachowanie „s” przez dodanie odpowiednich atrybutów (*flags*):

g	- każdy poszukiwany ciąg jest zastępowany w wierszu.
i	- przypadek poszukiwanego ciągu jest ignorowany.
p	- wiersz z poszukiwanym ciągiem jest wyprowadzany. Jeżeli

występuje razem z opcją „-n” - tylko pasujące wiersze są wyprowadzane.

```
geeko@da51:~ > sed -n 's/dol/XXXXX/p' loremipsum.txt
Lorem ipsum XXXXXor sit amet,
ut lacreet XXXXXore magna aliguam erat volutpat.
Duis autem XXXXXor in hendrerit
vel illum XXXXXore eu feugiat
```

w *nazwa_pliku* - wiersze spełniające warunek są zapisywane do pliku tak, jak:

```
sed 's/dol/XXXXX/w output.txt' loremipsum.txt
```

s - ciąg jest traktowany jako pojedynczy wiersz. Jest możliwe wtedy szukanie znaku „nowego wiersza” („\n”)

Można łączyć atrybuty, na przykład:

```
sed 's/dol/XXXXX/gi' loremipsum.txt
```

Czasami potrzebny jest wzorzec do porównania wyrażeń określonych w części *string* polecenia **sed**. Mamy do dyspozycji dziewięć rejestrów, w których możemy przechowywać wzorce. Należy odpowiednio oznaczyć (nawiasami \(...\)) wyrażenie, które chcemy zachować. Można odwołać się odpowiednio do dziewięciu wzorców w wierszu oznaczając je odpowiednio: \1 do \9.

```
geeko@da51:~ > sed  's/\([a-c]\)/X\1X/g' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET


Lorem ipsum dolor sit XaXmet,
XcXonseXcXtetuer XaXdipisXcXing elit,
sed diem nonummy niXbXh euismodtinXcXidunt
```

Wyjaśnienie. Wyrażenie regularne wyłapuje wszystkie „a”, „b” oraz „c” i dołącza do nich „X” z przodu i z tyłu.

- „**n” (next)** sprawdzany jest wiersz za wzorcem.

- „**h” (hold), „g”, „G” , „x”** bieżący wiersz jest wczytywany z **sed** do bufora (*pattern space*). Następnie jest przekształcany i wysyłany do kanału wyjścia. Za pomocą polecenia „**h**” możemy kopiować zawartość bufora *pattern space* do innego bufora, tzw. *holding buffer*. Za pomocą „**G**” można wstawić zawartość *holding buffer* do bieżącego wiersza.

```
geeko@da51:~ > sed '1h;3G' loremipsum.txt
```

```
LOREM IPSUM DOLOR SIT AMET

Lorem ipsum dolor sit amet,
LOREM IPSUM DOLOR SIT AMET
consectetuer adipiscing elit,
```

Wyjaśnienie. Pierwszy wiersz kopiowany jest do bufora a następnie wstawiany za wierszem trzecim.

Polecenie „**g**” zastępuje bieżący wiersz zawartością bufora:

```
geeko@da51:~ > sed '1h;3g' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET

LOREM IPSUM DOLOR SIT AMET
consectetuer adipiscing elit,
sed diem nonummy nibh euismodtincidunt
```

Polecenie „**x**” zamienia zawartość bufora *pattern space* oraz bufora *holding buffer*.

```
geeko@da51:~ > sed '1h;2x;4G' loremipsum.txt
LOREM IPSUM DOLOR SIT AMET
LOREM IPSUM DOLOR SIT AMET
Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,

sed diem nonummy nibh euismodtincidunt
ut lacreet dolore magna aliguam erat volutpat.
```

Wyjaśnienie. Pierwsze polecenie „1h” kopiuje wiersz pliku loremipsum.txt do bufora holding buffer. „2x'' zamienia zawartość drugiego wiersza (pusty wiersz) z zawartością bufora holding buffer. Zawartość bufora (pusty wiersz) jest wstawiany za wierszem czwartym, jako wynik polecenia „4G”.

Można zapisać całość lub część poleceń sed w zewnętrznym pliku i wywołać poleceniem sed z opcją „**-f**”:

```
sed -f commands.txt loremipsum.txt
```

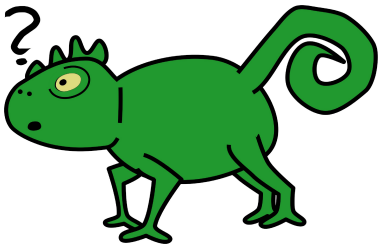
Zewnętrzny plik musi być zgodny z poniższymi zasadami:

1. Wiersz zaczynający się znakiem „#” jest komentarzem i nie będzie interpretowany;
2. Przed i za poleceniem dopuszczalne są dowolne wcięcia (spacje);
3. Przy zapisie więcej niż jednego polecenia w wierszu, należy je oddzielić średnikami.

Skrypt będzie wykonany dla każdego wiersza pliku tekstowego.

Ćwiczenie. Używanie edytora sed – część IV

Napisz skrypt sed, który zastępuje „dolor” przez „XXXXX” oraz każde „o” przez „O” - w pliku loremipsum.txt. Wszystkie wiersze zaczynające się od „Duis” powinny być usunięte. Wyjście powinno być zapisane w pliku scriptlorem.txt.



Przykłady omawiane w książce mogą wydawać się bardzo proste, tym niemniej można używać **sed** w programach bash'a do przekształcania plików tekstowych. Operacje dodawania i usuwania elementów pliku tekstowego są przydatne np. przy opracowywaniu automatycznie tworzonych plików logów.