

połączenia 1

1. Utworzyć w programie VIM dokument tekstowy z tekstem piosenki (ona tu jest i tańczy dla mnie...)



```
pasternak@B2020: ~/Pulpit/vim
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.

Moja dziewczyna patrzy często w oczy me
i nie ukrywam sprawia to przyjemność.
Jak ją kocham tylko moje serce wie,
gdy mnie całuje, oddałbym nie jedno.

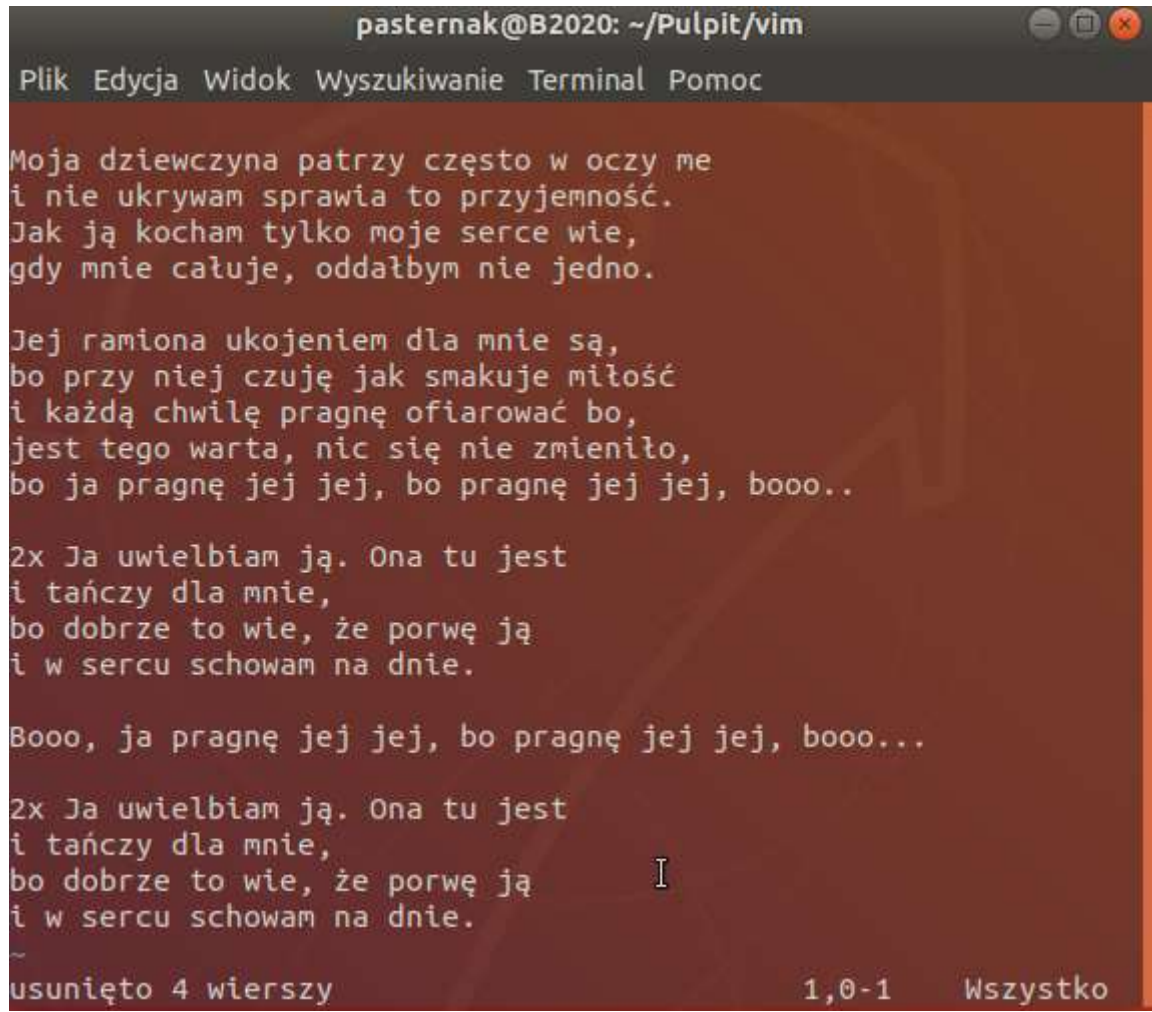
Jej ramiona ukojeniem dla mnie są,
bo przy niej czuję jak smakuje miłość
i każdą chwilę pragnę ofiarować bo,
jest tego warta, nic się nie zmieniło,
bo ja pragnę jej jej, bo pragnę jej jej, booo..

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.

Booo, ja pragnę jej jej, bo pragnę jej jej, booo...

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.
~
~
~
~
"piosenka" 27L, 766C 23,0-1 Wszystko
```

2. Usunąć 4 wiersze następnie jeszcze 7 wyraz



```
pasternak@B2020: ~/Pulpit/vim
Plik Edycja Widok Wyszukiwanie Terminal Pomoc

Moja dziewczyna patrzy często w oczy me
i nie ukrywam sprawia to przyjemność.
Jak ją kocham tylko moje serce wie,
gdy mnie całuje, oddałbym nie jedno.

Jej ramiona ukojeniem dla mnie są,
bo przy niej czuję jak smakuje miłość
i każdą chwilę pragnę ofiarować bo,
jest tego warta, nic się nie zmieniło,
bo ja pragnę jej jej, bo pragnę jej jej, booo..

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.

Booo, ja pragnę jej jej, bo pragnę jej jej, booo...

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.
~
usunięto 4 wierszy          1,0-1    Wszystko
```

i nie ukrywam sprawia to przyjemność.
Jak ją kocham tylko moje serce wie,
gdy mnie całuje, oddałbym nie jedno.

Jej ramiona ukojeniem dla mnie są,
bo przy niej czuję jak smakuje miłość
i każdą chwilę pragnę ofiarować bo,
jest tego warta, nic się nie zmieniło,
bo ja pragnę jej jej, bo pragnę jej jej, booo..

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.

Booo, ja pragnę jej jej, bo pragnę jej jej, booo...

2x Ja uwielbiam ją. Ona tu jest
i tańczy dla mnie,
bo dobrze to wie, że porwę ją
i w sercu schowam na dnie.

~

1 1 zmiana; prz... 13 sekund temu 2,0-1 Wszystko d7w

3. Zmienić wielkość liter (z dużych na małe i odwrotnie) 5 szt

JĄ UWIELBIAM JĄ, ONA TU JEST
I TAŃCZY DLA MNIE
BO DOBRZE TO WIE, ŻE PORWĘ JĄ
I W SERCU SCHOWAM NA DNIE, BO...

trzeba wejść do trybu wizualnego

("v"), zaznaczyć tekst (szczątki albo "hjl"), a następnie użyć polecenia g~

<https://leniwy.eu/news,5,Podstawy-edytora-Vim-skroty-klawiszowe,2.html>

JĄ UWIELBIAM JĄ, ONA TU JEST
I TAŃCZY DLA MNIE
BO DOBRZE TO WIE, ŻE PORWĘ JĄ
I W SERCU SCHOWAM NA DNIE, BO...
JĄ UWIELBIAM JĄ, ONA TU JEST

5g~ (5 linijek)

JĄ UWIELBIAM JĄ, ONA tu jest

g~5w (5 wyrazów)

4. Skopiować tekst „Ja uwielbiam ją” za pomocą polecenia a następnie skopiowanie go 10 razy



```
pasternak@B2020: ~/Pulpit
Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dnie, bo...

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dnie

Moja dziewczyna patrzy często w oczy me
I nie ukrywam, sprawia to przyjemność
Jak ją kocham, tylko moje serce wie
Gdy mnie całuje, oddałbym niejedno

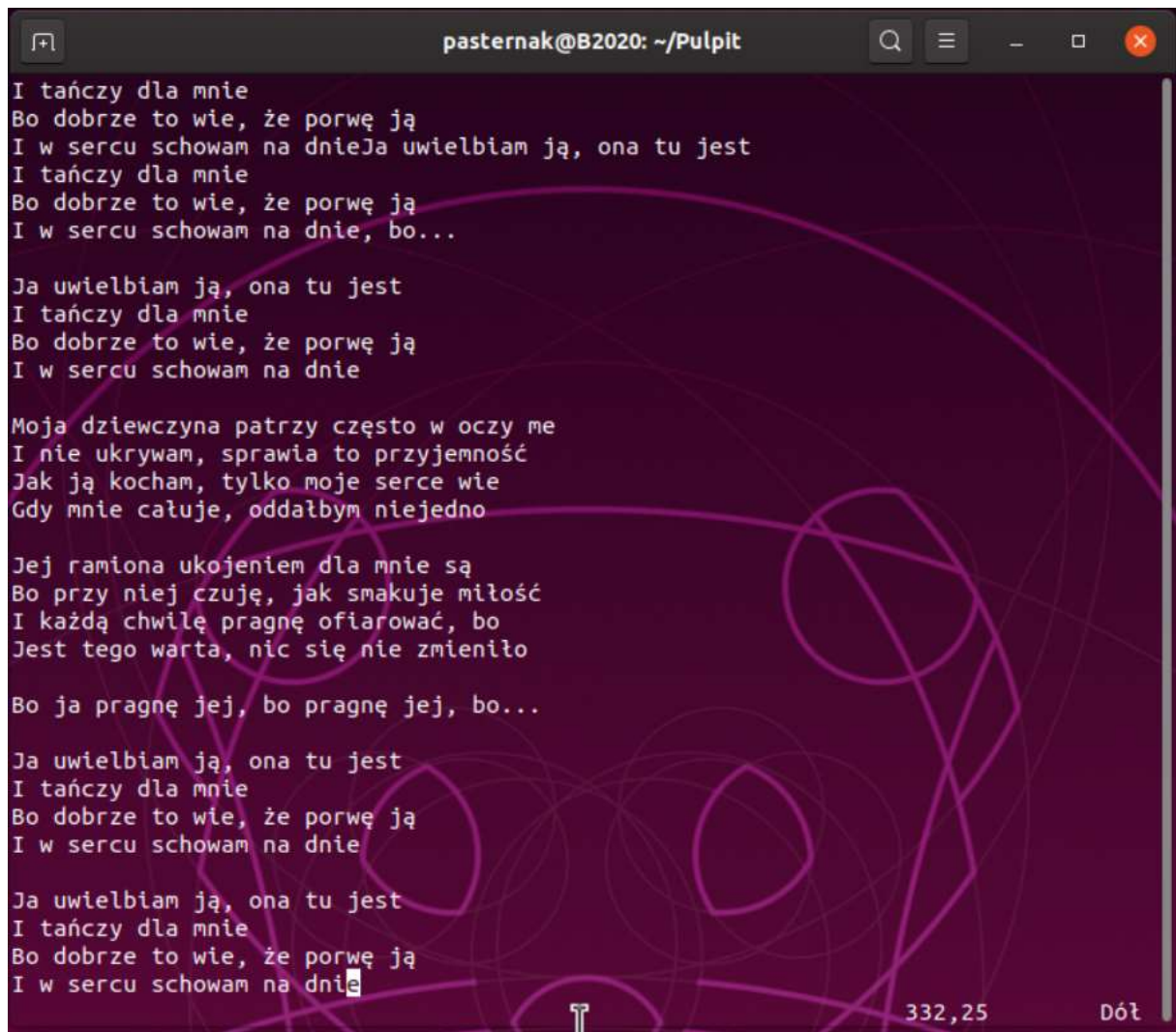
Jej ramiona ukojeniem dla mnie są
Bo przy niej czuję, jak smakuje miłość
I każdą chwilę pragnę ofiarować, bo
Jest tego warta, nic się nie zmieniło

Bo ja pragnę jej, bo pragnę jej, bo...

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dnie

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dnie
~
~
31 lines yanked
1,1 Wszystko
```

zaznaczyć tekst, a następnie polecenie “y”



```
pasternak@B2020: ~/Pulpit

I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dzieńJa uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dzień, bo...

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dzień

Moja dziewczyna patrzy często w oczy me
I nie ukrywam, sprawia to przyjemność
Jak ją kocham, tylko moje serce wie
Gdy mnie całuje, oddałbym niejedno

Jej ramiona ukojeniem dla mnie są
Bo przy niej czuję, jak smakuje miłość
I każdą chwilę pragnę ofiarować, bo
Jest tego warta, nic się nie zmieniło

Bo ja pragnę jej, bo pragnę jej, bo...

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dzień

Ja uwielbiam ją, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę ją
I w sercu schowam na dzień

332,25      Dół
```

a następnie wkleić "10p"

5. Słowo „ją” zamienić na „placki”

Ja uwielbiam placki, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę placki
I w sercu schowam na dnie, bo...

Ja uwielbiam placki, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę placki
I w sercu schowam na dnie

Moja dziewczyna patrzy często w oczy me
I nie ukrywam, sprawia to przyjemność
Jak placki kocham, tylko moje serce wie
Gdy mnie całuje, oddałbym niejedno

Jej ramiona ukojeniem dla mnie są
Bo przy niej czuję, jak smakuje miłość
I każdą chwilę pragnę ofiarować, bo
Jest tego warta, nic się nie zmieniło

Bo ja pragnę jej, bo pragnę jej, bo...

Ja uwielbiam placki, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę placki
I w sercu schowam na dnie

Ja uwielbiam placki, ona tu jest
I tańczy dla mnie
Bo dobrze to wie, że porwę placki
I w sercu schowam na dnie
Ja uwielbiam placki, ona tu jest
I tańczy dla mnie

można użyć polecenia

%s/tekst1/tekst2 (%s/ją/placki)

6. Oznaczyć słowo placki w dokumencie (wyświetl efekt i polecenie)

placki,

polecenie "v"

7. Posortować cały tekst

Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki
Bo ja pragnę jej, bo pragnę jej, bo...
Bo przy niej czuję, jak smakuje miłość
Gdy mnie całuje, oddałbym niejedno
I każdą chwilę pragnę ofiarować, bo
I nie ukrywam, sprawia to przyjemność
I tańczy dla mnie
I tańczy dla mnie
I tańczy dla mnie
I tańczy dla mnie
I w sercu schowam na dnie
I w sercu schowam na dnie
I w sercu schowam na dnie
I w sercu schowam na dnie, bo...
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
Jak placki kocham, tylko moje serce wie
Jej ramiona ukojeniem dla mnie są
Jest tego warta, nic się nie zmieniło
Moja dziewczyna patrzy często w oczy me

%!sort

8. Posortować tekst odwrotnie

Moja dziewczyna patrzy często w oczy me
Jest tego warta, nic się nie zmieniło
Jej ramiona ukojeniem dla mnie są
Jak placki kocham, tylko moje serce wie
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
Ja uwielbiam placki, ona tu jest
I w sercu schowam na dnie, bo...
I w sercu schowam na dnie
I w sercu schowam na dnie
I w sercu schowam na dnie
I tańczy dla mnie
I tańczy dla mnie
I tańczy dla mnie
I tańczy dla mnie
I nie ukrywam, sprawia to przyjemność
I każdą chwilę pragnę ofiarować, bo
Gdy mnie całuje, oddałbym niejedno
Bo przy niej czuję, jak smakuje miłość
Bo ja pragnę jej, bo pragnę jej, bo...
Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki
Bo dobrze to wie, że porwę placki

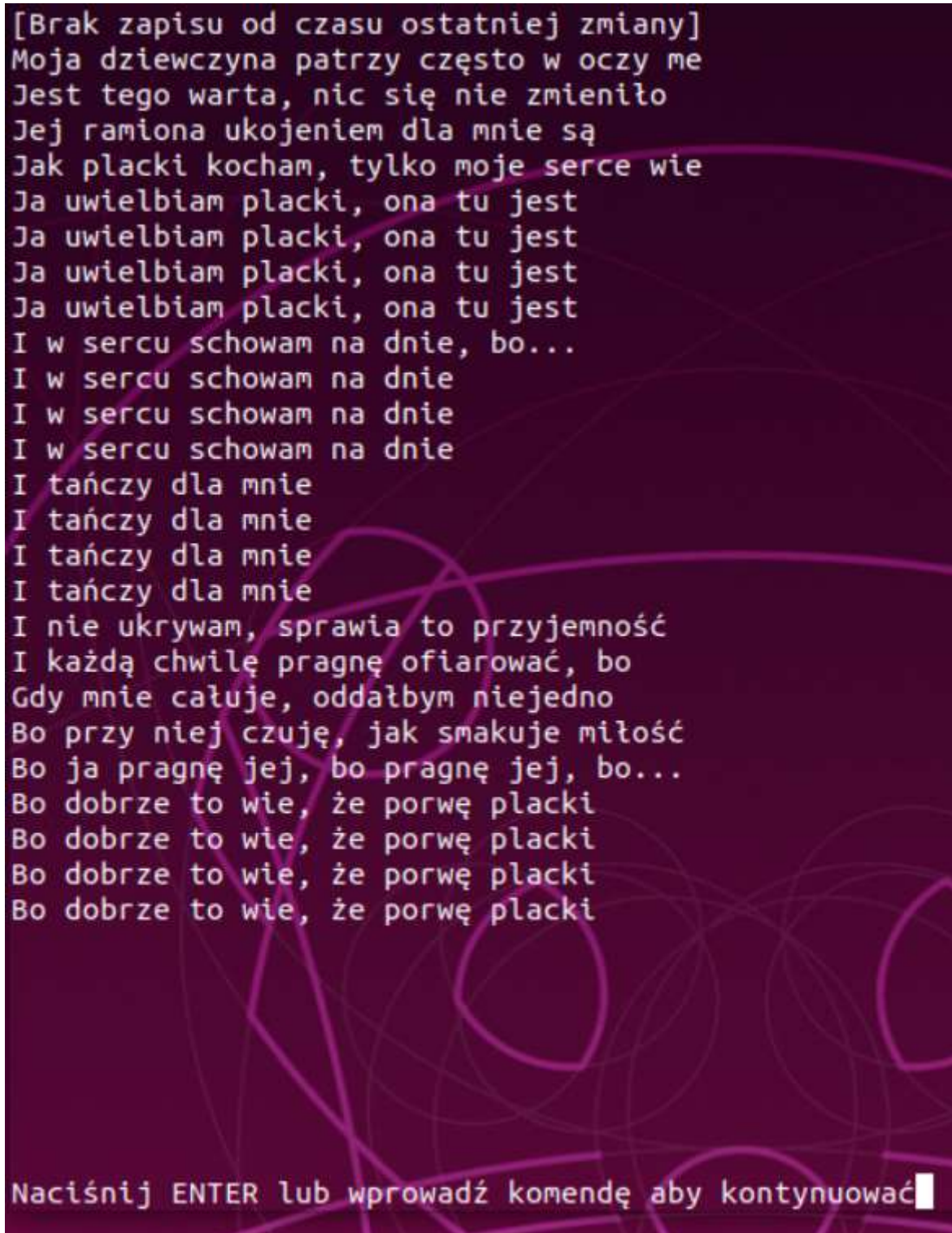
%!sort-r

9. Zapisać cały dokument pod swoim nazwiskiem na pulpicie (pokaż działanie)

"Pasternak" [Nowy] 31L, 827C zapisano

:w Pasternak

10. Wczytać tekst za pomocą terminalu i wyświetlenie zawartości



!cat Pasternak

1. Czym jest VIM ?

Vim (skrót od ang. vi improved) – wieloplatformowy klon edytora tekstu vi, napisany przez holenderskiego programistę Bramę Moolenaar. Vim należy do grupy wolnego oprogramowania o otwartym kodzie źródłowym. Pierwsza wersja została wydana w 1991 roku.

[Vim](#)

2. Czym jest VI ?

vi – edytor modalny, popularny w systemach Unix. Jego autorem jest Bill Joy. Nazwa pochodzi od angielskiego wyrazu visual.

Edytor vi w praktyce został wyparty przez klony, takie jak Vim, nvi lub elvis, które oferują bogatsze możliwości. Pozostaje jednym z cennych narzędzi administratora ze względu na powszechność i jednolitość implementacji.

[vi \(program\)](#)

3. Jakie występują tryby pracy programu i jak się je uruchamia?

występuje 7 trybów podstawowych i 7 dodatkowych.

Vim has seven BASIC modes:	
	Normal Normal-mode command-mode
Normal mode	In Normal mode you can enter all the normal editor commands. If you start the editor you are in this mode (unless you have set the 'insertmode' option, see below). This is also known as command mode.
Visual mode	This is like Normal mode, but the movement commands extend a highlighted area. When a non-movement command is used, it is executed for the highlighted area. See Visual-mode . If the 'showmode' option is on "-- VISUAL --" is shown at the bottom of the window.
Select mode	This looks most like the MS-Windows selection mode. Typing a printable character deletes the selection and starts Insert mode. See Select-mode . If the 'showmode' option is on "-- SELECT --" is shown at the bottom of the window.
Insert mode	In Insert mode the text you type is inserted into the buffer. See Insert-mode . If the 'showmode' option is on "-- INSERT --" is shown at the bottom of the window.
Command-line mode Cmdline mode	In Command-line mode (also called Cmdline mode) you can enter one line of text at the bottom of the window. This is for the Ex commands, ":", the pattern search commands, "?" and "/", and the filter command, "!". Cmdline-mode
Ex mode	Like Command-line mode, but after entering a command you remain in Ex mode. Very limited editing of the command line. Ex-mode
Terminal-Job mode	Interacting with a job in a terminal window. Typed keys go to the job and the job output is displayed in the terminal window. See terminal about how to switch to other modes.

sposób przełączania się między nimi ↓

		TO mode						
		Normal	Visual	Select	Insert	Replace	Cmd-line	Ex
FROM mode								
Normal			v V ^V	*4	*1	R gR	: / ? !	Q
Visual		*2		^G	c C	--	:	--
Select		*5	^O ^G		*6	--	--	--
Insert	I	<Esc>	--	--		<Insert>	--	--
Replace		<Esc>	--	--	<Insert>		--	--
Command-line		*3	--	--	:start	--		--
Ex		:vi	--	--	--	--	--	
		-- not possible						

4. Jakie są sposoby nawigacji po między znakami, wyrazami, wierszami?

```

~~~~~
Lekcja 1.1.: PORUSZANIE SIĘ KURSOREM
~~~~~

** By wykonać ruch kursorem, wciśnij h, j, k, l jak pokazano. **

      ^
      k
    < h   l >
      j
      v

Wskazówka: h jest po lewej
            l jest po prawej
            j wygląda jak strzałka w dół

1. Poruszaj kursorem dopóki nie będziesz pewien, że pamiętasz polecenia.
2. Trzymaj j tak długo aż będzie się powtarzał.
   Teraz wiesz jak dojść do następnej lekcji.
3. Używając strzałki w dół przejdź do następnej lekcji.

Uwaga: Jeśli nie jesteś pewien czegoś co wpisałeś, wciśnij <ESC>, by wrócić do
trybu Normal. Wtedy powtórz polecenie.

Uwaga: Klawisze kursora także powinny działać, ale używając hjkl będziesz
w stanie poruszać się o wiele szybciej, jak się tylko przyzwyczaisz.

```

5. Jak skopiować fragment tekstu i wklejać go dowolnie ?

6. Jak uruchomić podręcznik w programie VIM 

- a. Uruchomić Tutorial i wypełnić go w całości

```
Lekcja 1.3.: EDYCJA TEKSTU - KASOWANIE

** Wciśnij x aby usunąć znak pod kursorem. **

1. Przenieś kursor do linii poniżej oznaczonej --->.
2. By poprawić błędy, naprowadź kursor na znak do usunięcia.
3. Wciśnij x aby usunąć niechciany znak.
4. Powtarzaj kroki 2. do 4. dopóki zdanie nie jest poprawne.
---> Krowa przeskoczyła przez księżyc.
5. Teraz, kiedy zdanie jest poprawione, przejdź do Lekcji 1.4.

UWAGA: Ucz się przez ćwiczenie, nie wkuwanie.
```

```
Lekcja 1.4.: EDYCJA TEKSTU - INSERT (wprowadzanie)

** Wciśnij i aby wstawić tekst. **

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->.
2. Aby poprawić pierwszy wiersz, ustaw kursor na pierwszym znaku PO tym,
   gdzie tekst ma być wstawiony.
3. Wciśnij i a następnie wpisz konieczne poprawki.
4. Po poprawieniu błędu wciśnij <ESC>, by wrócić do trybu Normal.
   Powtarzaj kroki 2. do 4., aby poprawić całe zdanie.
---> W tej linii brakuje trochę tekstu.
---> W tej linii brakuje trochę tekstu.

5. Kiedy czujesz się swobodnie wstawiając tekst, przejdź do
   podsumowania poniżej.
```

Lekcja 1.5.: EDYCJA TEKSTU - APPENDING (dodawanie)

**** Wciśnij A by dodać tekst. ****

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->. Nie ma znaczenia, który to będzie znak.

2. Wciśnij A i wpisz odpowiednie dodatki.

3. Kiedy tekst został dodany, wciśnij <ESC> i wróć do trybu Normalnego.

4. Przenieś kursor do drugiej linii oznaczonej ---> i powtórz kroki 2. i 3., aby poprawić zdanie.

---> Brakuje tu trochę tekstu.

Brakuje tu trochę tekstu.

---> Tu też trochę brakuje. █

Tu też trochę brakuje.

5. Kiedy już utrwaliłeś ćwiczenie, przejdź do lekcji 1.6.

cześć,
właśnie nauczyłem się korzystać z edytor vim :)

teraz sprawdź czy da się zapisać zmiany

1200 █

cześć,
właśnie nauczyłem się korzystać z edytor vim :)

teraz sprawdź czy da się zapisać zmiany

"test" 7L, 112C

Lekcja 2.5.: UŻYCIE MNOŻNIKA, BY WIĘCEJ USUNĄĆ

**** Wpisanie liczby z operatorem powtarza go odpowiednią ilość razy. ****

W wyżej wspomnianej kombinacji operatora usuwania i ruchu podaj mnożnik przed ruchem, by więcej usunąć:

`d liczba ruch`

1. Przenieś kursor do pierwszego wyrazu KAPITALIKAMI w linii zaznaczonej --->.
2. Wpisz `2dw` aby usunąć dwa wyrazy KAPITALIKAMI.
3. Powtarzaj kroki 1. i 2. z innymi mnożnikami, aby usunąć kolejne wyrazy KAPITALIKAMI jednym poleceniem

---> ta linia wyrazów została oczyszczona.

UWAGA: Mnożnik pomiędzy operatorem `d` i ruchem działa podobnie do ruchu bez operatora.

Lekcja 2.6.: OPEROWANIE NA LINIACH

**** Wpisz `dd` aby usunąć całą linię. ****

Z powodu częstości usuwania całych linii, projektanci Vi zdecydowali, że będzie łatwiej wpisać dwa razy `d` aby usunąć linię.

1. Przenieś kursor do drugiego zdania z wierszyka poniżej.
2. Wpisz `dd` aby usunąć wiersz.
3. Teraz przenieś się do czwartego wiersza.
4. Wpisz `2dd` aby usunąć dwa wiersze.

---> 1) Róże są czerwone,
---> 3) Fiołki są niebieskie,
---> 6) Cukier jest słodki,
---> 7) I ty też.

Lekcja 2.7.: POLECENIE UNDO (cofnij)

**** Wciśnij u aby cofnąć skutki ostatniego polecenia.
U zaś, by cofnąć skutki dla całej linii. ****

1. Przenieś kursor do zdania poniżej oznaczonego ---> i umieść go na pierwszym błędzie.
2. Wpisz x aby usunąć pierwszy niechciany znak.
3. Teraz wciśnij u aby cofnąć skutki ostatniego polecenia.
4. Tym razem popraw wszystkie błędy w linii używając polecenia x .
5. Teraz wciśnij wielkie U aby przywrócić linię do oryginalnego stanu.
6. Teraz wciśnij u kilka razy, by cofnąć U i poprzednie polecenia.
7. Teraz wpisz CTRL-R (trzymaj równocześnie wciśnięte klawisze CTRL i R) kilka razy, by cofnąć cofnięcia.

---> Popraw błędy w tej linii zamień je przez cofnij.

8. To są bardzo pożyteczne polecenia.

Lekcja 2.7.: POLECENIE UNDO (cofnij)

**** Wciśnij u aby cofnąć skutki ostatniego polecenia.
U zaś, by cofnąć skutki dla całej linii. ****

1. Przenieś kursor do zdania poniżej oznaczonego ---> i umieść go na pierwszym błędzie.
2. Wpisz x aby usunąć pierwszy niechciany znak.
3. Teraz wciśnij u aby cofnąć skutki ostatniego polecenia.
4. Tym razem popraw wszystkie błędy w linii używając polecenia x .
5. Teraz wciśnij wielkie U aby przywrócić linię do oryginalnego stanu.
6. Teraz wciśnij u kilka razy, by cofnąć U i poprzednie polecenia.
7. Teraz wpisz CTRL-R (trzymaj równocześnie wciśnięte klawisze CTRL i R) kilka razy, by cofnąć cofnięcia.

---> Popraw błędy w tej linii i zamień je przez cofnij.

8. To są bardzo pożyteczne polecenia.

Przejdź teraz do podsumowania Lekcji 2.
Już w miejscu najnowszej zmiany

Lekcja 3.1.: POLECENIE PUT (wstaw)

**** Wpisz p by wstawić ostatnie usunięcia za kursorem. ****

1. Przenieś kursor do pierwszej linii ---> poniżej.
2. Wpisz dd aby usunąć linię i przechować ją w rejestrze Vima.
3. Przenieś kursor do linii c), POWYŻEJ tej, gdzie usunięta linia powinna się znajdować.
4. Wciśnij p by wstawić linię poniżej kursora.
5. Powtarzaj kroki 2. do 4. aż znajdą się w odpowiednim porządku.

---> a) Na górze róże,
---> b) Na dole fiołki,
---> c) A my się kochamy,
█---> d) Jak dwa aniołki.

Lekcja 3.2.: POLECENIE REPLACE (zastąp)

**** Wpisz rx aby zastąpić znak pod kursorem na x . ****

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->
2. Ustaw kursor na pierwszym błędzie.
3. Wpisz r a potem znak jaki powinien go zastąpić.
4. Powtarzaj kroki 2. i 3. dopóki pierwsza linia nie będzie taka, jak druga.

---> Kiedy ten wiersz był wstukiwany, ktoś wcisnął perę złych klawiszy!
---> Kiedy ten wiersz był wstukiwany, ktoś wcisnął parę złych klawiszy!

5. Teraz czas na Lekcję 3.3.

Lekcja 3.3.: OPERATOR CHANGE (zmień)

**** By zmienić do końca wyrazu, wpisz ce . ****

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->.
2. Umieść kursor na u w lunos.
3. Wpisz ce i popraw wyraz (w tym wypadku wstaw inia).
4. Wciśnij <ESC> i przejdź do następnej planowanej zmiany.
5. Powtarzaj kroki 3. i 4. dopóki pierwsze zdanie nie będzie takie same, jak drugie.

---> Ta linia ma parę słów, które trzeba zmienić używając polecenia zmien.

---> Ta linia ma parę słów, które trzeba zmienić używając polecenia zmien.

Zauważ, że ce nie tylko zamienia wyraz, ale także zmienia tryb na Insert (wprowadzanie).

Lekcja 3.4.: WIĘCEJ ZMIAN UŻYWAJĄC c

**** Polecenie change używa takich samych ruchów, jak delete. ****

1. Operator change działa tak samo, jak delete. Format wygląda tak:

c [liczba] ruch

2. Ruchy są także takie same, np.: w (wyraz), \$ (koniec linii) etc.
3. Przenieś się do pierwszej linii poniżej oznaczonej --->
4. Ustaw kursor na pierwszym błędzie.
5. Wpisz c\$, popraw koniec wiersza i wciśnij <ESC>.

---> Koniec tego wiersza musi być poprawiony używając polecenia c\$.

---> Koniec tego wiersza musi być poprawiony używając polecenia c\$.

UWAGA: Możesz używać <BS> aby poprawiać błędy w czasie pisania.

Lekcja 4.2.: POLECENIE SZUKAJ

**** Wpisz / a następnie wyrażenie, aby je znaleźć. ****

1. W trybie Normal wpisz /. Zauważ, że znak ten oraz kursor pojawiają się na dole ekranu tak samo, jak polecenie : .
2. Teraz wpisz blond<ENTER> . To jest słowo, którego chcesz szukać.
3. By szukać tej samej frazy ponownie, po prostu wciśnij n .
Aby szukać tej frazy w przeciwnym, kierunku wciśnij N .
4. Jeśli chcesz szukać frazy do tyłu, użyj polecenia ? zamiast / .
5. Aby wrócić gdzie byłeś, wciśnij CTRL-O. Powtarzaj, by wrócić dalej. CTRL-I idzie do przodu.

Uwaga: 'błond' to nie jest metoda, by przeliterować błąd; 'blond' to błąd.

Uwaga: Kiedy szukanie osiągnie koniec pliku, będzie kontynuowane od początku /blond

Lekcja 4.4.: POLECENIE SUBSTITUTE (zamiana)

**** Wpisz :s/stary/nowy/g aby zamienić 'stary' na 'nowy'. ****

1. Przenieś kursor do linii poniżej oznaczonej --->.
 2. Wpisz :s/czaas/czas<ENTER> . Zauważ, że to polecenie zmienia tylko pierwsze wystąpienie 'czaas' w linii.
 3. Teraz wpisz :s/czaas/czas/g . Dodane g oznacza zmianę (substytucję) globalnie w całej linii. Zmienia wszystkie wystąpienia 'czaas' w linii.
- > Najlepszy czas na zobaczenie najładniejszych kwiatów to czas wiosny.
4. Aby zmienić wszystkie wystąpienia łańcucha znaków pomiędzy dwoma liniami, wpisz: :#,#s/nowe/nowe/g gdzie #,# są numerami linii ograniczających region, gdzie ma nastąpić zamiana.
- wpisz :%s/nowe/nowe/g by zmienić wszystkie wystąpienia w całym pliku.
wpisz :%s/stare/nowe/gc by zmienić wszystkie wystąpienia w całym pliku, prosząc o potwierdzenie za każdym razem.

zamień na nowe (y/n/a/q/l/^E/^Y)?

Lekcja 5.1.: JAK WYKONAĆ POLECENIA ZEWNĘTRZNE?

**** Wpisz `:!` a następnie zewnętrzne polecenie, by je wykonać. ****

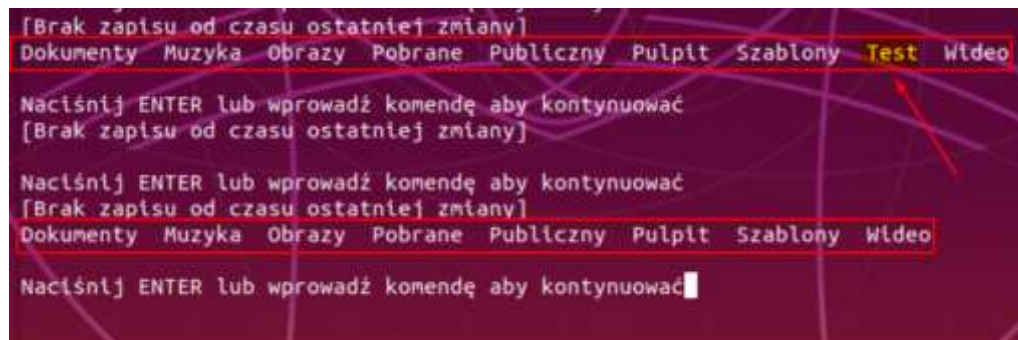
1. Wpisz znajome polecenie `:` by ustawić kursor na dole ekranu. To pozwala na wprowadzenie komendy linii poleceń.
2. Teraz wstaw `!` (wykrzyknik). To umożliwi Ci wykonanie dowolnego zewnętrznego polecenia powłoki.
3. Jako przykład wpisz `ls` za `!` a następnie wciśnij `<ENTER>`. To polecenie pokaże spis plików w Twoim katalogu, tak jakbyś był przy znaku zachęty powłoki. Możesz też użyć `!dir` jeśli `ls` nie działa.

Uwaga: W ten sposób można wykonać wszystkie polecenia powłoki.
Uwaga: Wszystkie polecenia `:` muszą być zakończone `<ENTER>`.
Od tego momentu nie zawsze będziemy o tym wspominać.

`!dir`

[Brak zapisu od czasu ostatniej zmiany]
Dokumenty Muzyka Obrazy Pobrane Publiczny Pulpit Szablony Wideo

Naciśnij ENTER lub wprowadź komendę aby kontynuować



Lekcja 5.3.: WYBRANIE TEKSTU DO ZAPISU

**** By zachować część pliku, wpisz v w ruch :w NAZWA_PLIKU ****

1. Przenieś kursor do tego wiersza.
2. Wciśnij v i przenieś kursor do punktu 5. Zauważ, że tekst został podświetlony.
3. Wciśnij znak : . Na dole ekranu pojawi się : '<,'> .
4. Wpisz w TEST , gdzie TEST to nazwa pliku, który jeszcze nie istnieje. Upewnij się, że widzisz : '<,'>w TEST zanim wciśniesz Enter.
5. Vim zapisze wybrane linie do pliku TEST. Użyj :!dir lub :!ls , żeby to zobaczyć. Jeszcze go nie usuwaj! Użyjemy go w następnej lekcji.

UWAGA: Wciśnięcie v zaczyna tryb wizualny. Możesz poruszać kursorem, by zmienić rozmiary zaznaczenia. Możesz też użyć operatora, by zrobić coś z tekstem. Na przykład d usuwa tekst.

Lekcja 5.4.: WSTAWIANIE I ŁĄCZENIE PLIKÓW

**** By wstawić zawartość pliku, wpisz :r NAZWA_PLIKU ****

1. Przenieś kursor do tego wiersza.
 2. Wciśnij v i przenieś kursor do punktu 5. Zauważ, że tekst został podświetlony.
 3. Wciśnij znak : . Na dole ekranu pojawi się : '<,'> .
 4. Wpisz w TEST , gdzie TEST to nazwa pliku, który jeszcze nie istnieje. Upewnij się, że widzisz : '<,'>w TEST zanim wciśniesz Enter.
 5. Vim zapisze wybrane linie do pliku TEST. Użyj :!dir lub :!ls , żeby to zobaczyć. Jeszcze go nie usuwaj! Użyjemy go w następnej lekcji.
1. Umieść kursor tuż powyżej tej linii.

UWAGA: Po wykonaniu kroku 2. zobaczysz tekst z Lekcji 5.3. Potem przejdź do DOŁU, by zobaczyć ponownie tę lekcję.

2. Teraz wczytaj plik TEST używając polecenia :r TEST , gdzie TEST jest nazwą pliku. Wczytany plik jest umieszczony poniżej linii z kursorem.
3. By sprawdzić czy plik został wczytany, cofnij kursor i zobacz, że teraz są dwie kopie Lekcji 5.3., oryginał i kopia z pliku.

UWAGA: Możesz też wczytać wyjście zewnętrznego polecenia. Na przykład :r !ls wczytuje wyjście polecenia ls i umieszcza je pod poniżej kursora.

Lekcja 6.1.: POLECENIE OPEN (otwórz)

**** Wpisz o by otworzyć linię poniżej kursora i przenieść się do trybu Insert (wprowadzanie). ****

1. Przenieś kursor do linii poniżej oznaczonej --->.
2. Wpisz o (małe), by otworzyć linię PONIŻEJ kursora i przenieść się do trybu Insert (wprowadzanie).
3. Wpisz trochę tekstu i wciśnij <ESC> by wyjść z trybu Insert (wprowadzanie)

---> Po wciśnięciu o kursor znajdzie się w otwartej linii w trybie Insert. trochę tekstu.....

4. By otworzyć linię POWYŻEJ kursora, wciśnij wielkie O zamiast małego o . Wypróbuj to na linii poniżej.

Trochę więcej tekstu.....

---> Otwórz linię powyżej wciskając SHIFT-O gdy kursor będzie na tej linii.

Lekcja 6.2.: POLECENIE APPEND (dodaj)

**** Wpisz a by dodać tekst ZA kursorem. ****

1. Przenieś kursor do początku pierwszej linii poniżej oznaczonej --->
2. Wciskaj e dopóki kursor nie będzie na końcu li .
3. Wpisz a (małe), aby dodać tekst ZA znakiem pod kursorem.
4. Dokończ wyraz tak, jak w linii poniżej. Wciśnij <ESC> aby opuścić tryb Insert.
5. Użyj e by przejść do kolejnego niedokończonego wyrazu i powtarzaj kroki 3. i 4.

---> Ta linia pozwoli Ci ćwiczyć dodawanie tekstu do końca linii.

---> Ta linia pozwoli Ci ćwiczyć dodawanie tekstu do końca linii.

Uwaga: a , i oraz A prowadzą do trybu Insert, jedyną różnicą jest miejsce, gdzie nowe znaki będą dodawane.

Lekcja 6.3.: INNA WERSJA REPLACE (zamiana)

**** Wpisz wielkie R by zamienić więcej niż jeden znak. ****

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->. Przenieś kursor do pierwszego xxx .
2. Wciśnij R i wpisz numer poniżej w drugiej linii, tak, że zastąpi on xxx.
3. Wciśnij <ESC> by opuścić tryb Replace. Zauważ, że reszta linii pozostaje niezmieniona.
5. Powtarzaj kroki by wymienić wszystkie xxx.

---> Dodanie 123 do 456 daje 579.

---> Dodanie 123 do 456 daje 579.

UWAGA: Tryb Replace jest jak tryb Insert, ale każdy znak usuwa istniejący znak.

Lekcja 6.4.: KOPIOWANIE I WKLEJANIE TEKSTU

**** użyj operatora y aby skopiować tekst i p aby go wkleić ****

1. Przejdź do linii oznaczonej ---> i umieść kursor za "a)".
2. Wejdź w tryb Wizualny v i przenieś kursor na początek "pierwszy".
3. Wciśnij y aby kopiować (yankować) podświetlony tekst.
4. Przenieś kursor do końca następnej linii: j\$
5. Wciśnij p aby wkleić (wpakować) tekst. Dodaj: a drugi<ESC> .
6. Użyj trybu Wizualnego, aby wybrać " element.", yankuj go y , przejdź do końca następnej linii j\$ i upakuj tam tekst z p .

---> a) to jest pierwszy element.

b) to jest drugi element.

Uwaga: możesz użyć y jako operatora; yw kopiuje jeden wyraz.

```

Lekcja 6.5.: USTAWIANIE OPCJI

** Ustawianie opcji tak, by szukaj lub substytucja ignorowały wielkość liter **

1. Szukaj 'ignore' wpisując:      /ignore<ENTER>
   Powtórz szukanie kilka razy naciskając klawisz n .

2. Ustaw opcję 'ic' (Ignore case -- ignoruj wielkość liter) poprzez
   wpisanie:                      :set ic

3. Teraz szukaj 'ignore' ponownie wciskając: n
   Zauważ, że ignore i IGNORE także są teraz znalezione.

4. Ustaw opcje 'hlsearch' i 'incsearch':      :set hls is

5. Teraz wprowadź polecenie szukaj ponownie i zobacz co się zdarzy:
   /ignore<ENTER>

6. Aby wyłączyć ignorowanie wielkości liter:  :set noic

Uwaga: Aby usunąć podświetlanie dopasowań, wpisz:      :nohlsearch
Uwaga: Aby ignorować wielkość liter dla jednego wyszukiwania: /ignore\c<ENTER>
/IGNORE

*help.txt*      For Vim version 8.1.  Last change: 2019 Jan 17

                VIM - main help file

Move around:    Use the cursor keys, or "h" to go left,      k
                "j" to go down, "k" to go up, "l" to go right. h l
Close this window: Use ":q<Enter>".                          j
Get out of Vim:  Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag (e.g. |bars|) and hit CTRL-].
With the mouse:  ":set mouse=a" to enable the mouse (in xterm or GUI).
                Double-click the left mouse button on a tag, e.g. |bars|.
Jump back:       Type CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the |:help| command.
Prepend something to specify the context: *help-context*

                WHAT          PREPEND    EXAMPLE
Normal mode command          :help x
Visual mode command          v_         :help v_u
Insert mode command          i_         :help i_<Esc>
Command-line command         :         :help :quit
Command-line editing         c_         :help c_<Del>
Vim command argument         -         :help -r
Option                       '         :help 'textwidth'
Regular expression           /         :help /[

See |help-summary| for more contexts and an explanation.

help.txt [Pomoc][R0]

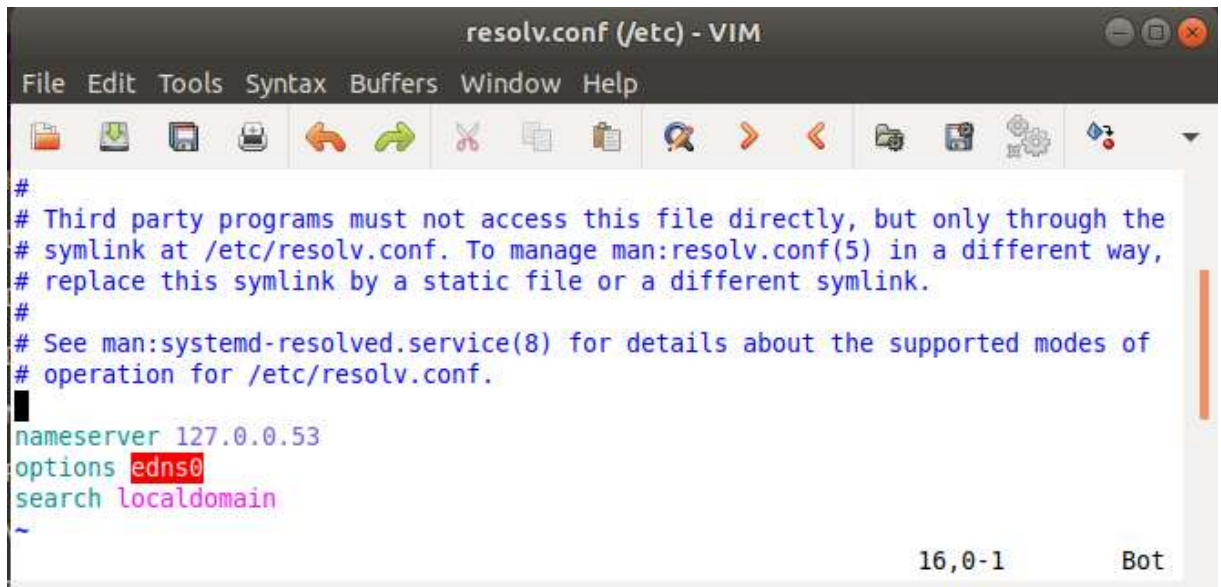
```

7. Jakie są inne edytory tekstu w Linuxie
 - a. Wbudowane

- Nano Editor
- Vim
- Vi

b. Zainstalować min 5 dodatkowych

- Gvim Editor

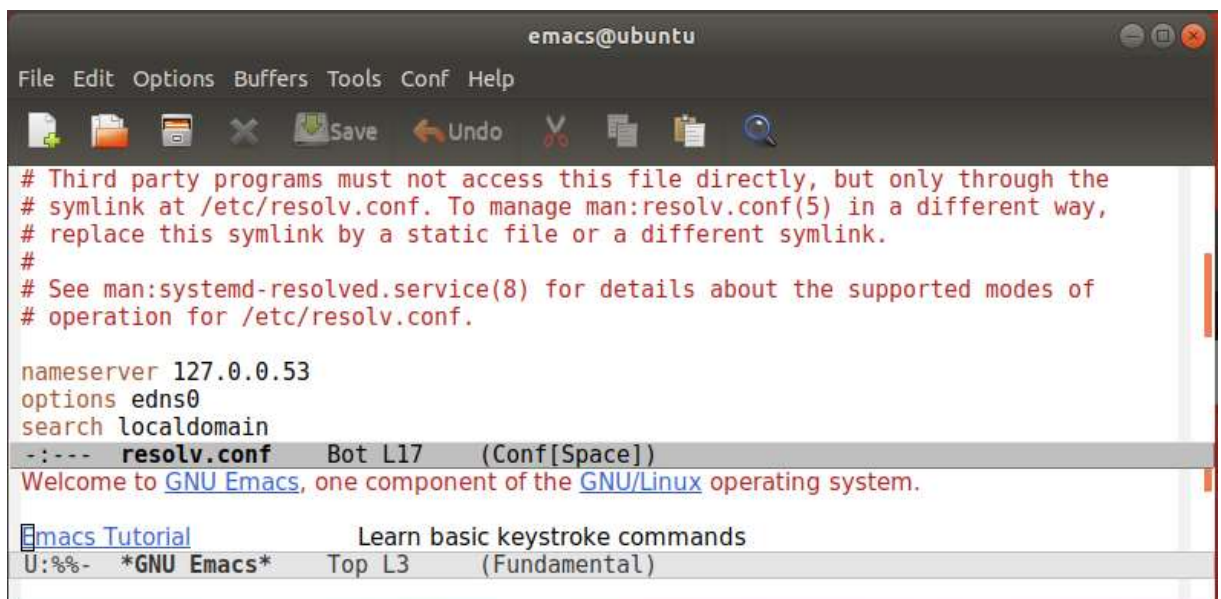


The screenshot shows the Vim editor window titled "resolv.conf (/etc) - VIM". The menu bar includes File, Edit, Tools, Syntax, Buffers, Window, and Help. The toolbar contains various icons for file operations and editing. The text content of the file is as follows:

```
#  
# Third party programs must not access this file directly, but only through the  
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,  
# replace this symlink by a static file or a different symlink.  
#  
# See man:systemd-resolved.service(8) for details about the supported modes of  
# operation for /etc/resolv.conf.  
nameserver 127.0.0.53  
options edns0  
search localdomain  
~
```

The status bar at the bottom right shows "16,0-1" and "Bot".

- Emacs



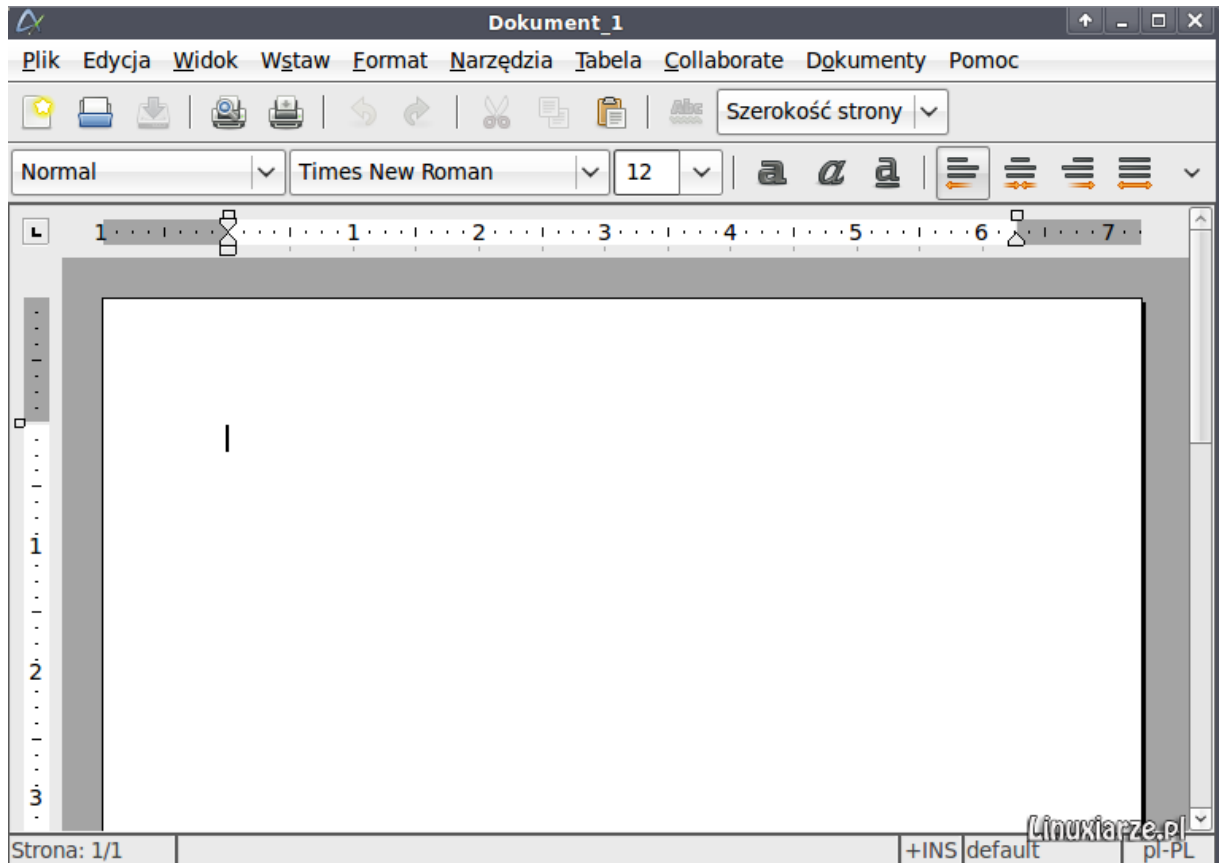
The screenshot shows the Emacs editor window titled "emacs@ubuntu". The menu bar includes File, Edit, Options, Buffers, Tools, Conf, and Help. The toolbar contains icons for file operations and editing. The text content of the file is as follows:

```
# Third party programs must not access this file directly, but only through the  
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,  
# replace this symlink by a static file or a different symlink.  
#  
# See man:systemd-resolved.service(8) for details about the supported modes of  
# operation for /etc/resolv.conf.  
nameserver 127.0.0.53  
options edns0  
search localdomain
```

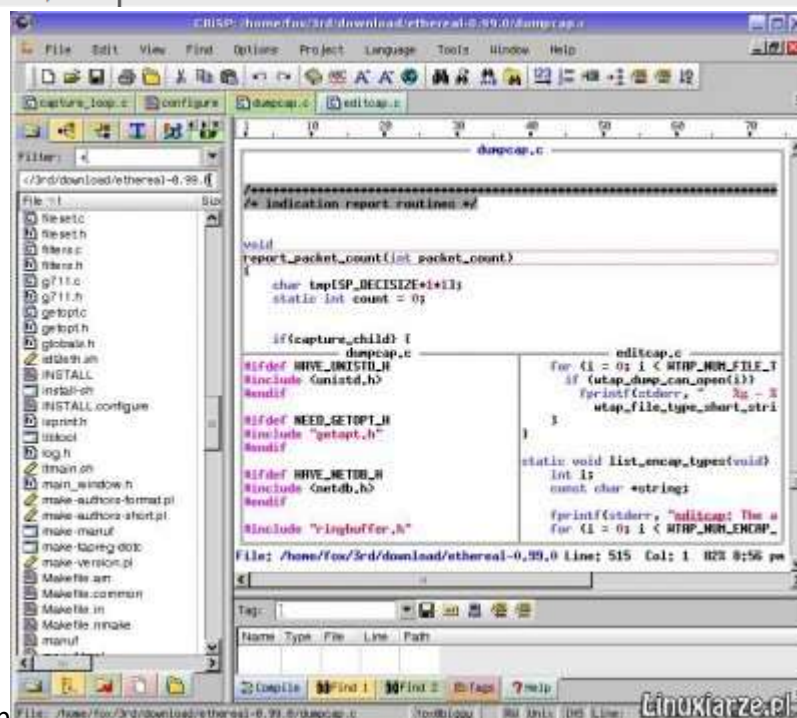
The status bar at the bottom shows the following information:

```
-:--- resolv.conf Bot L17 (Conf[Space])  
Welcome to GNU Emacs, one component of the GNU/Linux operating system.  
Emacs Tutorial Learn basic keystroke commands  
U:%%- *GNU Emacs* Top L3 (Fundamental)
```


- AbiWord



- Crisp



- e3

```
MicroEditor e3 v2.6.1-UTF8 GPL © 2000-04 A.Kleine <kleine@ak.sax.de>
Enter filename or leave with RETURN

Key bindings in WS mode:

Files:  ^KR Insert      ^KS Save      ^KX Save&Exit  ^KQ Abort&Exit
        ^KD Save&Load ^KP Pipe buffer thru 'sed'

Blocks: ^KB Start      ^KK End       ^KC Copy       ^KY Del
        ^KV Move      ^KW Write

Search: ^QF Find       ^L Repeat     ^QA Srch&Repl
Move:   ^E Up       ^X Down       ^S Left        ^D Right
        ^R Page Up  ^C Page Dn    ^W Scroll Up   ^Z Scroll Dn
Quick-  ^QE Wnd Top ^QX Wnd Bott  ^QS Home       ^QD End
-Move:  ^QR BOF     ^QC EOF       ^QB Blk Begin  ^QK Blk End
        ^F Next Word ^A Prev Word  ^QI Line#      ^QU Last Find

Delete: ^T Word       ^Y Line       ^H Left        ^G Chr
        ^QY Line End ^QDel, ^QH Line Beg

Other:  ^KM Set mode ^KN Numerics  ^KZ Suspend    ^U Undo

FILENAME:            Linuxiarze.pl
```

<http://leniwy.eu/news,5,Podstawy-edytora-Vim-skroty-klawiszowe.html>

http://computersun.pl/linux/podstawy/vim-podstawowy-edytor-tekstu-s_96.html

<http://vim.cba.pl/>

<http://www.openvim.com/tutorial.html>

<http://www.arturpyszczuk.pl/vimtutorial.html>

W tym tutorialu staram się podać najbardziej podstawowe polecenia używane w VIM'ie, w nagłówkach każdej sekcji w nawiasie podam w jakim trybie trzeba je wpisywać. Tryby te to: NORMAL, INSERT, VISUAL. Z trybu NORMAL do trybu INSERT przełączamy się poprzez i lub inne z wymienionych klawiszy w sekcji Edycja tekstu -> Dodawanie tekstu. Z trybu NORMAL do trybu VISUAL przełączamy się (oraz wyłączamy tryb VISUAL) poprzez klawisz v. Z trybu INSERT/VISUAL do trybu NORMAL przełączamy się za pomocą klawisza ESC.

ZAŁOŻENIA:

- Wiesz co to tryby VIM'a. Jeśli nie, to odsyłam Cię do Wikipedii, albo man vim :)
- Wszystkie polecenia zaczynające się od dwukropka (:) zakończone są naciśnięciem klawisza ENTER!
- W nawiasach < > wpisane są nazwy przycisków klawiatury! (np. <ENTER>, <HOME>, itd.)

Wszystkie te polecenia (oraz dużo dużo więcej) możesz znaleźć w dokumentacji, poprostu zebrałem je w jedną całość, w razie gdyby ktoś zapomniał i nie chciał mu się specjalnie szukać :)

1. Wchodzenie do VIM'a (przez konsolę):

- o vim - Otwiera VIM'a (bez skojarzonego pliku). Podczas zapisu trzeba będzie podać nazwę pliku, patrz :w nazwa poniżej.
- o vim nazwa.txt - Otwiera VIM'a wraz ze skojarzonym plikiem (użyteczne podczas pisania kodu np w C, php, html, itp., gdy wpiszymy od razu nazwę pliku i rozszerzenie (np 1.c) oraz gdy opcja kolorowania składni będzie włączona, VIM automatycznie będzie kolorował składnie dla danego typu plików), podczas zapisu wystarczy :w.
- o vim plik1 plik2 - Otwieranie z powłoki systemowej dwóch plików: plik1, plik2.
- o vim *.c - Otwieranie z powłoki systemowej wszystkich plików z rozszerzeniem .c.

2. Wychodzenie z VIM'a (-- NORMAL --):

- o :q - Zamyka plik pod warunkiem, że nie był zmodyfikowany, jeśli był, patrz niżej.
- o :q! - Zamyka plik bez zapisywania zmian.
- o :w - Zapisuje zmiany.
- o :w nazwa - Zapisuje zmiany do pliku o nazwie 'nazwa'.
- o :wq - Zapisuje zmiany i zamyka plik. Gdy edytujemy jeden plik, możemy zapisać i zakończyć używając polecenia :wq i włączyć drugi plik do edycji za pomocą vim plik2, gdzie plik2 to nazwa drugiego pliku. VIM oferuje inne, wygodniejsze rozwiązanie.
- o :vi plik2, gdzie plik2 to nazwa drugiego pliku. Włączamy to z poziomu vima, edytując plik pierwszy. Jeśli nie zapiszemy zmian w pliku pierwszym, VIM nas o tym poinformuje.
- o :vi! plik2, gdzie plik2 to nazwa drugiego pliku. Plik, który aktualnie edytujemy zostanie zamknięty, bez zapisywania ostatnio dokonanych zmian. Oraz zostanie włączony plik2 do edycji.
- o :e plik2 Działa dokładnie tam samo, jak powyższe polecenie :vi plik2
- o :view plik2 Działa prawie tak samo jak polecenia wyżej (:vi plik2, :e plik2) z tym, że plik2 zostanie otwarty tylko do odczytu.

3. Poruszanie się (-- NORMAL --):

- o h / strzałka w lewo (również -- INSERT --) - lewo
- o l / strzałka w prawo (również -- INSERT --) - prawo

- o j / strzałka w dół (również -- INSERT --) - dół
- o k / strzałka do góry (również -- INSERT --) - góra
- o O / <HOME> - Przesuwa kursor na początek
- o \$ / <END> - Przesuwa kursor na koniec
- o <^> - Przesuwa kursor na pierwszy nie-biały znak, np w zdaniu " Ala ma kota", kursor zostanie ustawiony na literę 'A'
- o w - Przesuwa kursor na początek następnego wyrazu
- o e - Przesuwa kursor na koniec następnego wyrazu
- o b - Przesuwa kursor na początek poprzedniego wyrazu
- o ge - Przesuwa kursor na koniec poprzedniego wyrazu
- o gg - Przesuwa kursor do pierwszego wiersza w pliku
- o G - Przesuwa kursor do ostatniego wiersza w pliku
- o :n - Przesuwa kursor do wiersza n - tego (:12 - Przesuwa kursor do wiersza 12).
- o nG - gdzie n to numer wiersza. Działanie takie samo jak :n (np 30G - przeniesie do 30 wiersza)
- o :\$ - To samo co G.
- o CTRL-G - Pokazuje położenie w pliku i status pliku
- o CTRL-U - Przesuwa ekran o połowę tekstu do góry.
- o CTRL-D - Przesuwa ekran o połowę tekstu w dół.
- o :next - Przechodzi do następnego pliku, pod warunkiem, że plik aktualny nie był zmodyfikowany, jeśli był, VIM pokaże ostrzeżenie.
- o :previous - Analogicznie do polecenia wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :wnext - Przechodzi do następnego pliku, oraz zapisuje aktualnie edytowany plik.
- o :wprevious - Analogicznie do polecenia wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :next! - Przechodzi do następnego pliku, nie zapisując zmian jakie zostały zrobione w aktualnym pliku.
- o :previous! - Analogicznie do polecenia wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :2next - Przesuwa się nie na następny a jeden dalej plik. Analogicznie istnieją opcję w tył, w przód z/bez zapisu.

- o :args - Pokazuje otwarte pliki, oraz w nawiasach kwadratowych, aktualnie edytowany plik, np. plik1.txt [plik2.txt] plik3.txt
- o :rewind - Przenosi do pierwszego otwartego pliku (zakładamy, że mamy otwartych kilka plików) nie ważne, który aktualnie plik edytujemy.
- o :first - To samo co wyżej.
- o :last - Przenosi do ostatniego otwartego pliku (zakładamy, że mamy otwartych kilka plików) nie ważne, który aktualnie plik edytujemy.

4. Mnożniki dla ruchu (-- NORMAL --):

- o Można zastosować mnożnik ruchu, który pozwoli przesunąć kursor o większą ilość wyrazów, np.
- o 2w - Przesuwa kursor na początek 2 wyrazu w prawo (licząc od pozycji kursora).
- o 2e - Przesuwa kursor na koniec 2 wyrazu w prawo (licząc od pozycji kursora).
- o 2b - Przesuwa kursor na początek 2 wyrazu w lewo (licząc od pozycji kursora).
- o 2ge - Przesuwa kursor na koniec 2 wyrazu w lewo (licząc od pozycji kursora). itp, itd.
- o 9k - Przesuwa kursor o 9 pozycji w górę (równoważne z: kkkkkkkkk)
- o 3j - Przesuwa kursor o 3 pozycje w dół (równoważne z: jjj)
- o 5l - Przesuwa kursor o 5 pozycji w prawo (równoważne z: lllll)
- o 2h - Przesuwa kursor o 2 pozycje w lewo (równoważne z: hh)

5. Edycja tekstu (-- NORMAL --):

- o Dodawanie tekstu:(-- NORMAL --)
- ❓ i - Wstawianie tekstu przed kursorem.
- ❓ a - Wstawianie tekstu za kursorem.
- ❓ A - Wstawianie tekstu na końcu linii.
- ❓ o - Wstawianie tekstu do nowej (następnej) linii.
- ❓ O - Wstawianie tekstu do nowej (poprzedniej) linii.

o Mnożniki używane podczas wstawiania tekstu są bardzo użyteczne. Np, jeśli chciałbyś wstawić 10 liter 'A' obok siebie, lub każda w nowym wierszu, oczywiście możesz to wpisywać ręcznie, ale VIM udostępnia mnożniki, tak więc, aby wpisać 10 liter 'A' szybciej, przejdź do trybu NORMAL oraz:

❓ 10iA<ESC> - Wstawienie 10 liter 'A' za kursorem

❓ 10oFraza<ESC> - Wstawienie 10 wyrazów 'Fraza', każdy w nowej linii. Zauważ, że po naciśnięciu '10o' VIM przechodzi w tryb INSERT, co za tym idzie, wszystko co wpiszesz, po naciśnięciu ESC zostanie skopiowane tyle razy ile podałeś w poleceniu (tutaj 10 razy).

o Kopiowanie i wklejanie tekstu (-- NORMAL -- / -- VISUAL --):

❓ y - Operator kopiowania, używać go można w trybie wizualnym (opisany na dole), lub z operatorami ruchu.

❓ yw - Kopiuje jeden wyraz

❓ y3w - Kopiuje 3 wyrazy

❓ :ny - Gdzie n to numer wiersza, np. :30y - Skopiuje wiersz 30.

❓ p - Operator wklejania.

❓ np - Gdzie n to wielokrotność kopiowanego tekstu, np. 10p - Wkleji 10 razy, to co zostało skopiowane.

o Łączenie linii (-- NORMAL --):

❓ J - Łaczy 2 linie (kursor trzeba ustawić na najwyższej). Pomiędzy liniami dodaje spacje

❓ 5J - Łaczy 5 linii, między każdą dodaje spacje.

o Usuwanie tekstu (-- NORMAL --):

❓ x / del - Usuwa znak pod kursorem.

❓ dw - Usunięcie jednego wyrazu do pierwszego znaku nowego wyrazu (innymi słowami: usuwa wyraz i spacje przed następnym wyrazem).

❓ de - Usunięcie jednego wyrazu do ostatniego znaku usuwanego wyrazu (innymi słowami: usuwa tylko wyraz a spacja przed następnym wyrazem zostaje).

❓ d\$ - Usunięcie tekstu od kursora do końca linii.

❓ dd - Usunięcie całego wiersza niezależnie od położenia kursora.

o Mnożniki: (-- NORMAL --)

? Mnożniki dla edycji tekstu działają właściwie tak samo, jak mnożniki ruchu.

? d2e - Usunięcie 2 wyrazów.

? d2w - Usunięcie 2 wyrazów (różnice pomiędzy de i dw zostały opisane wyżej).

? d3d - Usunięcie 3 linii, itp. itd.

o Oznaczenie (Mark) (-- NORMAL --):

? ma - 'm' to operator zaznaczenia, 'a' to tak jakby miejsce zaznaczenia. Można wybrać zamiast 'a' litery od a do z. Operator zaznaczenia przydaje się np do usuwania znacznej ilości bloków tekstu.

? 'a - Po tym jak zaznaczymy pewne miejsce przez operator 'm', to odwołujemy się do niego poprzez 'a (apostrof a, jak wyżej wspomniano, zamiast a, może być inna litera alfabetu).

? d'a - Dzięki takiej konstrukcji, VIM usuwa wszystko od miejsca zaznaczenia (ma) do miejsca, w którym znajduje się kursor

? y'a - Dzięki takiej konstrukcji, VIM kopiuje wszystko od miejsca zaznaczenia (ma) do miejsca, w którym znajduje się kursor

? :marks - Pokazuje listę wszystkich zaznaczeń.

? :marks args - Zamiast 'args' możesz wpisać konkretne zaznaczenie, np. a, b, itp., które chcesz zobaczyć. Lub wpisać to polecenie tak jak jest, razem z args.

o Cofanie (-- NORMAL --):

? u - Cofa skutki ostatniego polecenia

? U - Cofa skutki dla całej linii

? CTRL-R - Cofa cofnięcia, innymi słowy: Cofnij/Do przodu.

o Wstawianie (-- NORMAL --):

? p - Wstawia ostatnio usunięte / skopiowane znaki.

o Zamienianie (-- NORMAL --):

? r - Po naciśnięciu tego klawisza VIM czeka, aż podasz znak, na który ma zamienić wskazywany przez kursor znak.

? 5ra - Zamienia 5 znaków na prawo od aktualnej pozycji kursora na literę 'a'.

? R - Zamienia więcej niż jeden znak, działa na takiej samej zasadzie, jakbyś pisał z wciśniętym insertem (przycisk na klawiaturze, nie tryb pracy VIM'a).

- ❓ ce - Usuwa wyraz od miejsca wskazywanego przez kursor do końca wyrazu, przechodzi w tryb INSERT i czeka na wprowadzenie wyrazu.
- ❓ cw - Prawie to samo co ce, różnica jest taka sama jak pomiędzy de i dw.
- ❓ c\$ - Usuwa linie od miejsca wskazywanego przez kursor do końca linii, przechodzi w tryb INSERT i czeka na wprowadzenie wyrazu.
- ❓ Mnożniki również tu występują. Zasada jest analogiczna: cnw, gdzie n to liczba całkowita (1, 2, ..., n), zamiast w, może być e, \$, itp.
- ❓ c2e - Usuwa 2 wyrazy od miejsca wskazywanego przez kursor do końca 2 wyrazu i przechodzi w tryb INSERT, czeka na wprowadzenie nowych wyrazów.
- ❓ ~ - (tylda) Zamienia znak nad którym znajduje się kursor z dużego na mały i odwrotnie.
- ❓ 10~ - Zmienia wielkość 10 znaków na prawo od miejsca, w który znajduje się kursor.
- ❓ xp - Przydatne w sytuacji, gdy popełnisz literówkę, np. 'teh' zamiast 'the', ustaw kursor w słowie 'teh' na 'e' i wydaje to polecenie.
- o Szukaj (-- NORMAL --):
- ❓ /nazwa_blad<ENTER> - VIM szuka od początku pliku frazy: nazwa_blad
- ❓ ?nazwa_blad<ENTER> - VIM szuka od tyłu pliku frazy: nazwa_blad
- ❓ /<ENTER> - Znajduje kolejne wystąpienie słowa szukanego, w tym przypadku będzie to nazwa_blad
- ❓ /^word<ENTER> - Znajduje wyraz 'word' jeśli on zaczyna linię.
- ❓ /word\$<ENTER> - Znajduje wyraz 'word' jeśli on kończy linię.
- ❓ /c.m<ENTER> - Znajduje słowo zaczynające się na c, drugi znak dowolny a trzeci znak to m.
- ❓ /xxx\c<ENTER> - Wyszukuje frazę 'xxx' bez uwzględnienia wielkości liter (przełącznik \c).
- ❓ n - VIM szuka kolejnych wystąpień szukanej frazy od początku do końca.
- ❓ 10n - Szuka dziesiątego wystąpienia szukanej frazy w przód.
- ❓ N - VIM szuka kolejnych wystąpień szukanej frazy od końca do początku.
- ❓ 10N - Szuka dziesiątego wystąpienia szukanej frazy w tył
- ❓ /<UP> - <UP> - Strzałka do góry. Przeszukuje historie szukania.
- ❓ /<DOWN> - <DOWN> - Strzałka w dół. Przeszukuje historie szukania.
- ❓ CTRL-O - Przenosi kursor w poprzednie wystąpienie szukanej frazy (tam, gdzie byłeś)
- ❓ CTRL-I - Przenosi kursor w następne wystąpienie szukanej frazy (tam, gdzie byłeś)

- ❓ fx - Szuka znaku 'x' na prawo od kursora. Ustawia kursor na pierwszym znalezionym znaku 'x'.
- ❓ 5fa - Przeszukiwanie oraz ustawienie kursora na piątej literze 'a' występującej na prawo od miejsca kursora
- ❓ 10f(spacja) - Przeszukiwanie oraz ustawienie kursora na 10 spacji występującej na prawo od miejsca kursora
- ❓ Fi - Szuka znaku 'i' na lewo od kursora. Ustawia kursor na pierwszym znalezionym znaku 'i'
- ❓ 2Fo - Przeszukiwanie oraz ustawianie kursora na drugiej literze 'o' występującej na lewo od miejsca kursora
- ❓ to - Szuka znaku 'o' na prawo od kursora. Ustawia kursor na znaku poprzedzającym znak 'o'.
- ❓ To - Szuka znaku 'o' na lewo od kursora. Ustawia kursor na znaku poprzedzającym znak 'o'.
- o Szukanie brakujących nawiasów ((,) , [,] , { , })(-- NORMAL --):
- ❓ % - Szuka brakującego nawiasu, przenosi do miejsca, w którym występuje zamykający/otwierający brakujący nawias.
- o Substitute (znajdź i zamień)(-- NORMAL --):
- ❓ :s/xxx/yyy - Zamienia frazę 'xxx' na 'yyy' w linii, gdzie znajduje się kursor. Zauważ, że polecenie to zamieni tylko jedno wystąpienie frazy 'xxx', co jeśli chcemy zamienić wszystkie wystąpienia tejże frazy w tej linii? odpowiedź niżej.
- ❓ :s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii, w której znajduje się kursor.
- ❓ :s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w lini, w której znajduje się kursor, prosząc za każdym razem o potwierdzenie.
- ❓ :10s/xxx/yyy - Zamienia frazę 'xxx' na 'yyy' w linii nr. 10 (kursor nie musi być ustawiony w tej linii, polecenie to jest bardziej precyzyjne)
- ❓ :10s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii nr. 10
- ❓ :10s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii nr. 10, za każdym razem prosząc o potwierdzenie akcji.
- ❓ :%s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w całym pliku
- ❓ :%s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w całym pliku, lecz za każdym razem prosząc o potwierdzenie.
- ❓ :#s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' zawierającej się pomiędzy wierszami #, #.
- ❓ :#s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' zawierającej się pomiędzy wierszami #,# za każdym razem prosząc o potwierdzenie

o Polecenie zewnętrzne (-- NORMAL --):

❑ `!:nazwa_polecenia` - powłoka wykona polecenie 'nazwa_polecenia' (np. `!!ls -l` pokaże listę plików katalogu, w którym jesteśmy).

o Wybranie tekstu do zapisu (-- VISUAL --):

❑ `v` - Przełącza do trybu wizualnego. Możesz używać strzałek do zaznaczenia tekstu i jak już pewnie zobaczyłeś tekst jest podświetlony, ale również możesz używać operatorów ruchu: `w` (słowo), `e` (słowo), `$` (cała linia od kursora). Wciśnij `;`, a na dole pojawi się coś takiego: `:'<,>`. Wpisz za tym `'w nazwa'` (oczywiście bez `"`), gdzie nazwa to nieistniejącego pliku. Mniej więcej powinno wyglądać to tak: `:'<,>w nazwa`. Teraz, aby sprawdzić, czy zaznaczony tekst zapisał się do nowego pliku wystarczy wpisać `!cat nazwa`

o Filtrowanie (-- NORMAL --):

❑ `!20Gsort<ENTER>` - Jeśli ustawimy 20 wierszy, w każdym będzie jakaś liczba, to po wydaniu tego polecenia, VIM posortuje je rosnąco.

❑ `!Gsort<ENTER>` - Posortuje wiersze od miejsca, w którym znajduje się kursor do końca pliku.

❑ `!!ls -l` - Wstawi listę znajdujących się plików w katalogu. Dla użytkowników MS Windows: `!!dir`

❑ `!!date` - Wstawi datę systemową.

o Wstawienie i łączenie plików (-- NORMAL --):

❑ `:r nazwa` - Gdzie nazwa to nazwa pliku wstawianego. Można również wstawić zawartość polecenia zewnętrznego, tj. `:r !ls -l`

6. Ustawianie opcji (-- NORMAL --):

o `:set ic - ic` (Ignore Case) po ustawieniu tej opcji, wyrazy będą przeszukiwane bez względu na wielkość ich liter.

o `:set noic` - Wyłącza opcję `ic`.

o `:set hlsearch` - Włącza podświetlenie wyszukiwanego tekstu.

o `:set nohl` - Wyłącza podświetlenie wyszukiwanego tekstu.

o `:set incsearch` - Włącza częściowe dopasowanie tekstu podczas szukania.

o `:set noxxx` - Wyłącza opcję `xxx`

o `:set autowrite` - Włącza opcję autozapisywania.

o `:set noautowrite` - Wyłącza opcję autozapisywania.

7. Używanie tabulacji (-- NORMAL --)

- o << - (nawias trójkątny) Przesuwa linię w lewo, w której znajduje się kursor o wartość tabulacji.
- o >> - Przesuwa linię w prawo, w której znajduje się kursor o wartość tabulacji.
- o #,> - #,# - Numery wierszy. Przesuwa w prawo o wartość tabulacji wiersze podane jako argumenty.
- o #,<< - #,# - Numery wierszy. Przesuwa w lewo o 2 wartości tabulacji wiersze podane jako argumenty.

8. Używanie Makr (-- NORMAL --)

Posiadasz taką o to listę:

- o stdio.h
- o stdlib.h
- o sys/types.h
- o sys/stat.h
- o unistd.h
- o fcntl.h
- o errno.h

I chcesz, żeby wyglądała ona tak (za pomocą jednego polecenia):

- o #include <stdio.h>
- o #include <stdlib.h>
- o #include <sys/types.h>
- o #include <sys/stat.h>
- o #include <unistd.h>
- o #include <fcntl.h>
- o #include <errno.h>

Wykonuj pokolei wypunktowane poniżej polecenia, ale zanim to zrobisz przenieś kursor do pierwszego znaku w pierwszej linii:

- o qa - Zaczyna nagrywać makro, przypisując je literze a. Litera musi być z przedziału <a,z>
- o ^ - Przenosi na początek linii

- o `i#include <<ESC>` - Wstawia napis `#include <` na początku linii. (pamiętaj, że to `<ESC>` to jest przycisk na klawiaturze!)
- o `$` - Przechodzi na koniec linii
- o `a><ESC>` - Dodaje znak: `>` na końcu linii.
- o `j` - Przechodzi do następnej linii
- o `q` - Kończy nagrywanie makra

Teraz, żeby się odwołać do tego makra, przenieś kursor na pierwszy znak w pierwszej linii i w trybie normal wydaj polecenie: `@a`. Żeby to zrobić dla wszystkich naraz wpisz: `7@a`.

9. Dwuznaki (Digraphs) (`-- NORMAL --` / `-- INSERT --`):

- o `:digraphs` - Wyświetla listę dostępnych znaków
- o `CTRL-Kxy` - (Tryb INSERT), x, y - oznaczają znaki, które można odczytać z `:digraphs`. Dajmy na to przykład, żeby otrzymać znak: `Ł` (`C= Ł 1062`, Informacja z `:digraphs`) musimy nacisnąć `CTRL-K` i wpisać `C=` (dwa znaki: `C, =`)
- o `CTRL-KD%` - Analogicznie da inny znak, w tym wypadku ten znak to: `Љ`. (Informacje z `:digraphs`: `D% Љ 1026`)

10. Okna (`-- NORMAL --`)

- o `:split` - Polecenie to dzieli okno VIMa na 2 okna, w których edytuje ten sam plik.
- o `CTRL-Ww` / `CTRL-W CTRL-W` - Przełącza okna
- o `CTRL-Wj` - Przełącza z górnego okienka do dolnego
- o `CTRL-Wk` - Przełącza z dolnego okienka do górnego
- o `ZZ` / `:q` / `CTRL-Wc` - Zamyka jedno okienko
- o `:split plik2.txt` - Otwiera w drugim okienki plik: `plik2.txt`
- o `:15 split plik2.txt` - Otwiera `plik2.txt` w drugim oknie o rozmiarze 15 wierszy
- o `:new` - Prawie to samo co `split`, tylko, że otwiera nowy plik, w oknie.
- o `:10 new` - Otwiera nowy plik w oknie o rozmiarze 10 wierszy.
- o `:sview plik2.txt` - Otwiera `plik2.txt` w nowym oknie, tylko do odczytu.
- o `10sview plik2.txt` - Otwiera `plik2.txt` w nowym oknie o rozmiarze 10 wierszy, tylko do odczytu.
- o `10CTRL-W+` - Powiększa okno o 10 wierszy.
- o `10CTRL-W-` - Pomniejsza okno o 10 wierszy.

- o 12CTRL-W_ - Ustawia rozmiar okna na 12 wierszy.
- o CTRL-W_ - Ustawia okno na największą możliwość
- o CTRL-W= - Ustawia okna na takie same rozmiary (albo bardzo zbliżone)