```
BASIC vi COMMANDS:

Enter and exit

        i           Enter insert mode.
        R           Enter overwrite mode.
        <ESCAPE>    exit insert or overwrite mode.
        :q          exit vi.
        :wq         exit vi, saving file.
        :q!         exit vi, no save even if modifications done.
        ^G          prints out current line number and position in the file.

Inserting

        a           insert text after cursor.
        A           insert text after end of line.
        i           insert text before cursor.
        I           insert text at the begining of line.
        o           insert a line below current one.
        O           insert a line above current one.

Moving cursor

        nG          goto line n.
        G           goto end of file.
        0           goto beginnig of line.
        $           goto end of line.
        n|          cursor at column n.
        ^F          next page.
        ^B          previous page.
        ^D          next half page.
        ^U          previous half page.
        w           next word.
        b           previous word.
        e           end of word.
        'a          go to "a" marker line (see below for markers definition).

Modifying

        rx          replace single char by x.
        cw          replace current word by new text.
        C           replace from cursor to end of line by new text.
        J           fuse current line with the next.

Call to UNIX operating system

        :! command
                execute UNIX system command.
        :X,Y ! command
                execute  UNIX  system  command,  the  standard input and
                output  of  the command being the lines X to Y. Example:
                :12,$!sort will sort lines 12 to end of file inside your
                editing  session.  All  UNIX  system  commands  are thus
                available for text editing, formatting, etc!...


Deleting

                x           delete current character.
                X           delete previous character.
                dd          delete current line.
                ndd         delete n lines.
                dw          delete current word.
                D           delete from cursor to end of line.
        All deleted items are put into the scrap, i.e. can be pasted
        as described below.

Copy to scrap and paste scrap.

        yy          copy current line to scrap.
        nyy         copy n lines to scrap.
        p           paste scrap below current line.
        P           paste scrap above current line.
        :r tutu     insert the content of "tutu" file below cursor.
        :r! command insert the result of UNIX command below cursor.

Use of markers VERY USEFUL!...
```

```
        ma          put "a" marker at current line (one can use 26 markers: "a" to "z").
        y'a         copy all lines from the "a" marker to the current line to the scrap.
        d'a         delete all lines from the "a" marker to the current line to the scrap
.
        :'a,.w tutu write lines from "a" marker to the current line on file "tutu".

Line referencing

        One can reference lines on which a command will be executed
        by different manners; the generic form is

                :X,Y command, where X and Y can be either

                        - integers: they stand in that case for line NUMBERS
                        - 'a, 'b, ...: they stand in that case for markers
                        - the "$" character: they stand in that case for last line of the
 file
                        - the "." character: they stand in that case for current line

                If you type "%" instead of "X,Y", you stand for the hole file.

                All those manners can be used on the same line, like in the following:

                        :12,18w tutu
                                write from line number 12 to 18 on file tutu
                        :'a,.s/tutu/tata/g
                                replace all occurrences of "tutu" by "tata" from "a" mark
er to the current line.
                        :.,'b!sort
                                sort from current line to the "b" marker one,
                                and put the result at the same place in the editing sessi
on
                        :'d,$d
                                delete all lines from "d" marker to the end of file
                        :%>
                                indent the hole file in the right direction

Searching for text

        /tutu       search forward to the "tutu" pattern.
        ?tutu       search backward to the "tutu" pattern.
        n           repeat last search.
        N           repeat last search in reverse sense.
        :g/tutu/p   search all occurrences of the "tutu" pattern in the file and prints t
hem.
        :g/tutu/#   search all occurrences of the "tutu" pattern in the file and prints t
hem with line numbers.
        /tutu$      search forward to the "tutu" pattern at an end of line.
        /^tutu      search forward to the "tutu" pattern at the beginning of a line.
        %           locate matching (), {} or [].

Replacing text

        :X,Ys/tutu/tata/g   replace occurrences of "tutu" by "tata" from line X to Y.
        :X,Ys/tutu/tata/gc  idem but ask for confirmation for each replacing.
        :X,Y>               indent lines from X to Y in the right direction
        :X,Y<               indent lines from X to Y in the left direction


Undoing modifications

        u           undo modifications on current line.
        U           restore initial state of current line.
        .           repeat last modification command.

Multiple file editing

        vi f1 f2 ...
        :n          edit next file in the list.
        :rew        goto to first file in the list.
        :e file     edit new file in the current vi session.

Setting options:

                :set ic     ignore case in regular search.
                :set tabstop=6 tabstop are to be equivalent to 6 spaces.
                :set showmode prints out the mode (insert, overwrite, ...).
                :set nu     show line numbers.
```

```
        :set all    prints out all vi options.

These commands can be put in the $HOME/.exrc file, and then
will be active in each new vi session.
```