1 Using the visual editor `vi`

To create a file such as `factorial` on a Unix based computer, some familiarity with `vi` is necessary. Type `vi factorial`. You are now in command mode. Typing `a` allows one to enter text entry mode – any characters subsequently typed at the keyboard will be entered after the cursor. To finish text entry, hit the escape key. Then type `ZZ or :wq` to "write" the file.

Here is a short summary of vi commands.

- `dd` deletes the line the cursor is on; 2dd deletes two lines;
- `x` deletes the character under the cursor; 2x deletes two characters;
- `dw` deletes from the cursor to the start of the next word; 2dw deletes up to the start of the second next word;
- `dW` deletes up to the start of the next word, skipping punctuation;
- `cW` changes up to the start of the next word, skipping punctuation;
- `s` substitutes text for the character under the cursor;
- `S` substitutes text for the current line;
- `D (=d$)` deletes to the end of the current line;
- `d0` deletes from the beginning of the current line;
- `o` to insert text on a new line opened up after the current line;
- `O` to insert text on a new line opened up before the current line;
- `a` to insert text after the cursor;
- `A` to insert text at the end of the current line;
- `i` to insert text before the cursor;
- `I` to insert text before the first character on the current line;
- `h` to move left one character;
- `l` to move right one character;
- `j` to move down one line;

- `k` to move up one line;
- `w` to move the cursor to the start of the next word;
- `e` to move the cursor to the end of the word;
- `b` to move the cursor to the start of the previous word;
- `2Y` yanks two lines from the current line position; a copy of these lines can then be placed below a subsequent cursor position by typing `p`;
- `:i,jmk` places lines `i` to `j` after line `k`;
- `u` undoes the last change – a very important command!; repeated use of `u` in VIM is an indefinite *undo*, while repeating `CTRL-R` sequentially undoes the undoes;
- `U` undoes all changes to the current line;
- `.` repeats the last editing change – input, replacement or deletion;
- `:r filename` reads in a file called `filename` below the current line;
- `:3,12w temp` writes lines 3 to 12 into a file called `temp`;
- `:.w!  temp` writes the current line to a file called `temp`, overwriting a previous file of that name;
- `:.,.+3 w!  temp` writes the current line + two further lines to a file called `temp`, overwriting a previous file of that name;
- `CTRL g` displays the current line number;
- `/string` searches forward for the first occurence of `string`;
- `n` repeats the same string search in the same direction;
- `N` repeats the same string search in the opposite direction;
- `CTRL d or f` to move the window down;
- `CTRL b or u` to move the window up;
- `CTRL h` backspaces over characters while in insert mode;
- `:1,$s/searchstring/newstring/g` replaces all occurences of `searchstring` by `newstring`;

- `H` moves the cursor to the top of the screen;
- `L` moves the cursor to the bottom of the screen;
- `M` moves the cursor to the middle of the screen;
- `:w` to save the file, but remaining in `vi`;
- 3 >> shifts 3 lines to the right;
- 3 << shifts 3 lines to the left;
- >`%` when the cursor is placed on one of a pair of matching brackets, will shift the corresponding lines to the right;
- `nG` places the cursor at the start of line `n`;
- `CTRL ^` allows one to move to the previous file when editing multiple files;
- `2k` moves the cursor up 2 lines;
- `c2b` changes 2 words in a backwards direction;
- `"3p` puts the contents of buffer number 3 below the cursor; This works only for a deleted line. Words, or a portion of a line, are not saved in a buffer;
- `vi ‘fgrep -l string *html‘` locates all html files in the current directory which contain `string`;
- `z RETURN` moves current line to top of screen and scroll;
- `z.` moves current line to centre of screen and scroll;
- `z-` moves current line to bottom of screen and scroll;
- `200z` RETURN moves line 200 to top of screen;
- `` `0 `` takes you back to where you were when last exiting VIM;
- `:next` or `:n` takes you to next file when editing multiple files in VIM;
- `:previous` takes you to previous file when editing multiple files in VIM;
- `:args` lists the files when editing multiple files in VIM, with `[file]` denoting the current file.

To exploit the power of `vi`, the reader is urged to consult one of books:

- "A guide to vi", by Dan Sonnenschein (Prentice–Hall Software Series),
- "Visual editing on UNIX", by B. Srinivasan and K. Ranai (World Scientific),
- "Learning the vi editor", by Linda Lamb and Arnold Robbins (O'Reilly and Associates).
- "The ultimate guide to VI and EX text editors", Hewlett Packard, Benjamin Cummings.