Lab 1

Exercise 1.1: Moving around in Emacs

1. Typed emacs then C-x C-f search for exer1.html
2. C-s typed PUBLIC
3. C-s typed Laboratory
4. C-s typed self-referential
5. C-s typed arrow
6. C-e
7. C-a
8. C-s because it is the fastest way to find a word by typing.
9. No
10. C-x C-c

Exercise 1.2: Deleting text in Emacs

1. Type emacs then C-x C-f search for exer2.html
2. M-g g 18 C-k
3. C-s typed “DELETEME..” C-a C-k
4. C-s typed <https://en.wiki>.. C-a C-k
5. C-s < ! C-a C-k, C-s < ! C-a C-k C-x C-c

Exercise 1.3: Inserting text in Emacs

1. Type emacs then C-x C-f search for exer3.html
2. M-% Assignment 1 with Assignment 37 entered y y then q. q is exist query replace. Double y means 2 replacements.
3. M-% UTF-8 with US-ASCII entered y then q. q is exist query replace.
4. C-s </ol> C-a enter
5. C-x C-c and y.

Exercise 1.4: Other editing tasks in Emacs

1. cat exer2.html exer2.diff >exer4.html

1. Type emacs then C-x C-f search for exer4.html
2. M-> C-r C-SP C-e M-w M-g g 352 Enter C-e C-n C-y
3. M-> C-r <!--Enter C-SP C-u 7 C-f M-w M-g g 337 Enter C-s lines Enter C-y DEL DEL DEL

M-> C-r <!--Y Enter C-SP M-} M-w M-g g 300 C-u 3 C-n C-y DEL DEL DEL

M-> C-r <!--D Enter C-SP C-e M-w M-g g 90 Enter C-u 3 C-n 3 C-y DEL DEL DEL

M-> C-r <!--di Enter C-SP M-} M-w M-g g 15 Enter C-u 3 C-n C-y DEL DEL DEL

1. C-u C-\_
2. C-r </html C-e C-n <!-- M-> -->
3. M-< M-x count-matches Enter <ol> Enter

7 Occurrences

M-x replace-string Enter <ol> Enter <Ol> Enter

1. C-x C-c y

Exercise 1.5: Doing commands in Emacs

1. M-x make-directory Enter junk enter

2. C-x d junk Enter C-x b hello.c Enter C-x C-w Enter

3. M-x typed compile then DEL 8 times then typed gcc hello.c

4. C-x b hello.out C-u M-! ./a.out

5. M-< C-SP M-> M-w

Exercise 1.6: Running Elisp code

1. typed emacs then C-x b then \*scratch\* then enter

2. (random) C-j

3. (setq x (random)) C-j, (setq y (random)) C-j

4. (\* x y) C-j

5. M-: (\* x y) the same as number 4

6. No they are not. These random numbers are pseudo-random numbers and they are generated from a “seed”. The random function always generates the same sequence of numbers based on the “seed”.

7. I set random numbers for 20 variables, and I calculate (\* variable variable) for each pair of those which is 10 times, the outcomes is 30% wrong.