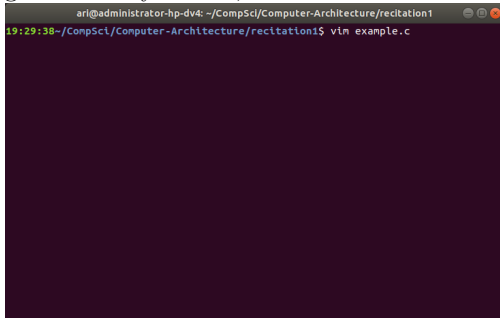


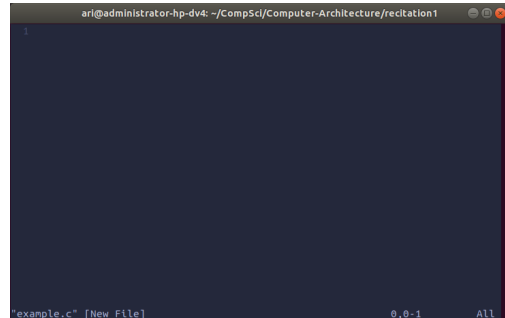
Vim — Example Usage

Ari Feiglin (ari.feiglin.biu@gmail.com)

1. Create a new blank file with vim. You are now in *normal mode*, where you can access other modes, navigate across your file, etc.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
19:29:38 ~/CompSci/Computer-Architecture/recitation1$ vim example.c
```



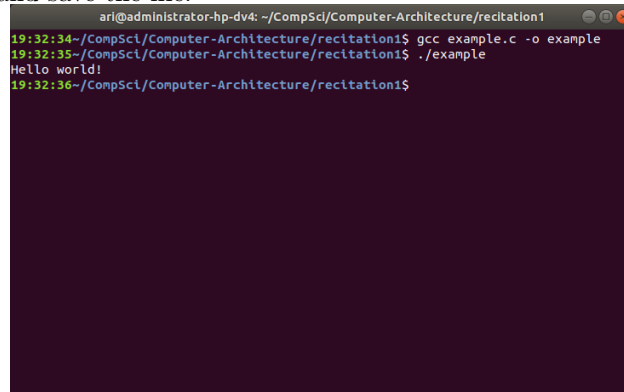
```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
1
"example.c" [New File] 0,0-1 All
```

2. Go into *insert mode* (by hitting the *i* key), and type text as you would normally in a text editor.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
1 #include <stdio.h>
2
3 int main(int argc, char** argv) {
4     puts("Hello world!");
5 }
6
"example.c" 6L, 83C 1,1 All
```

3. Go back into normal mode (ESC) and then into command mode (colon, :). Save the file with *w* (be sure to hit enter afterward). This will put you back into normal mode, go back into command mode and quit (*q*). These two commands can also be combined into one: *:wq*. Compile and run the C file to show that vim did indeed create and save the file.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
19:32:34 ~/CompSci/Computer-Architecture/recitation1$ gcc example.c -o example
19:32:35 ~/CompSci/Computer-Architecture/recitation1$ ./example
Hello world!
19:32:36 ~/CompSci/Computer-Architecture/recitation1$
```

4. Use vim to open the provided example file (foo.c).



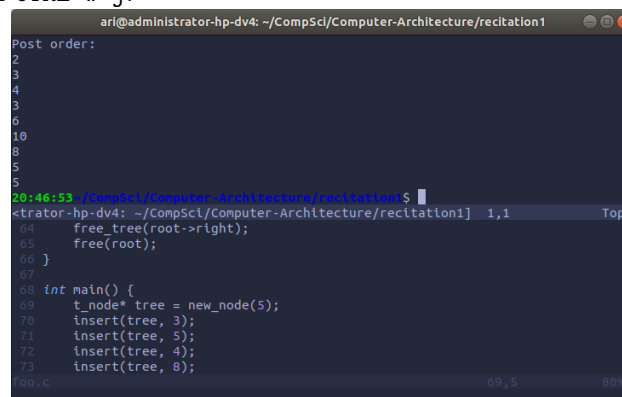
```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct s_node {
5     int val;
6     struct s_node* left;
7     struct s_node* right;
8 } t_node;
9
10 t_node* new_node(int val) {
11     t_node* root = malloc(sizeof(t_node));
12     if (root == NULL) {
13         perror("New node error");
14         exit(1);
15     }
16     root->left = NULL;
17     root->right = NULL;
18     root->val = val;
19     return root;
20 }
21
22 void insert(t_node* root, int val) {
23     foo.c" 86L, 1753C 1,1 Top
```

5. Move the cursor utilizing the h,j,k,l keys.
6. Go to the bottom G and top gg of the file.
7. Go to a specific line, eg. 69G.
8. Navigate to line 69, and delete a character using x, a word with dw, and the entire line with dd.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
58 }
59
60 void free_tree(t_node* root) {
61     if (root == NULL)
62         return;
63     free_tree(root->left);
64     free_tree(root->right);
65     free(root);
66 }
67
68 int main() {
69     Delete this text with dd
70     t_node* tree = new_node(5);
71     insert(tree, 3);
72     insert(tree, 5);
73     insert(tree, 4);
74     insert(tree, 8);
75     insert(tree, 3);
76     insert(tree, 2);
77     insert(tree, 6);
78     insert(tree, 10);
79
80     puts("Pre order: ");
81
82     foo.c" 69,5 86%
```

9. Demonstrate that the terminal can be accessed from within vim by :term and navigate to the terminal (to navigate between windows, first do CTRL-w, then a navigation key, h,j,k,l. In this case it will be CTRL-w k). Compile and run the program. (Remember to save any changes with w before running!) Navigate back to the editor with CTRL-w j.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
Post order:
2
3
4
3
6
10
8
5
5
20:46:53 ~/CompSci/Computer-Architecture/recitation1$
administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1] 1,1 Top
64     free_tree(root->right);
65     free(root);
66 }
67
68 int main() {
69     t_node* tree = new_node(5);
70     insert(tree, 3);
71     insert(tree, 5);
72     insert(tree, 4);
73     insert(tree, 8);
74
75     foo.c" 69,5 86%
```

10. Line 86 has a typo — it should be “Good bye!”. Navigate to line 86, to the correct column and fix this issue (navigate on top of the !, hit i, and insert an e).
11. Wherever you feel like it, show that you can go to the end of a line with \$, the beginning with 0, and to the first non-whitespace character with ^.

12. Navigate to line 70 and go into *visual mode* (linewise: V), select until line 78 and yank the contents using y. Then paste the contents above the cursor (the cursor will be brought back to line 70) using P. To paste the contents below the cursor, use p. Compile and run again.

```

ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
58 }
59
60 void free_tree(t_node* root) {
61     if (root == NULL)
62         return;
63     free_tree(root->left);
64     free_tree(root->right);
65     free(root);
66 }
67
68 int main() {
69     /* Delete this text */
70     t_node* tree = new_node(5);
71     insert(tree, 3);
72     insert(tree, 5);
73     insert(tree, 4);
74     insert(tree, 8);
75     insert(tree, 3);
76     insert(tree, 2);
77     insert(tree, 6);
78     insert(tree, 10);
79
80     puts("Pre order: ");
81 }
-- VISUAL LINE --
70,22 86%

```

```

ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
65     free(root);
66 }
67
68 int main() {
69     /* Delete this text */
70     t_node* tree = new_node(5);
71     insert(tree, 3);
72     insert(tree, 5);
73     insert(tree, 4);
74     insert(tree, 8);
75     insert(tree, 3);
76     insert(tree, 2);
77     insert(tree, 6);
78     insert(tree, 10);
79     insert(tree, 3);
80     insert(tree, 5);
81     insert(tree, 4);
82     insert(tree, 8);
83     insert(tree, 3);
84     insert(tree, 2);
85     insert(tree, 6);
86     insert(tree, 10);
87
88     puts("Pre order: ");
89 }
8 more lines
71,5 86%

```

```

ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
3
5
6
6
8
10
10
8
5
5
10:50:34 ~/CompSci/Computer-Architecture/recitation1$
<trator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1] 1,1 Top
73     insert(tree, 4);
74     insert(tree, 8);
75     insert(tree, 3);
76     insert(tree, 2);
77     insert(tree, 6);
78     insert(tree, 10);
79     insert(tree, 3);
80     insert(tree, 5);
81     insert(tree, 4);
82     insert(tree, 8);
foo.c
80,5 82%

```

13. Search for the pre-order function using the search command / (ie. do something like /pre_order). Navigate through the results with n (next) and N (previous).

```

ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
33     prev->right = new;
34 }
35
36 void print_in_order(t_node* root) {
37     if (root == NULL)
38         return;
39     print_in_order(root->left);
40     printf("%d\n", root->val);
41     print_in_order(root->right);
42 }
43
44 void print_pre_order(t_node* root) {
45     if (root == NULL)
46         return;
47     printf("%d\n", root->val);
48     print_pre_order(root->left);
49     print_pre_order(root->right);
50 }
51
52 void print_post_order(t_node* root) {
53     if (root == NULL)
54         return;
55     print_post_order(root->left);
56     print_post_order(root->right);
57 }
58
59 int main() {
60     /* Delete this text */
61     t_node* tree = new_node(5);
62     insert(tree, 3);
63     insert(tree, 5);
64     insert(tree, 4);
65     insert(tree, 8);
66     insert(tree, 3);
67     insert(tree, 2);
68     insert(tree, 6);
69     insert(tree, 10);
70     insert(tree, 3);
71     insert(tree, 5);
72     insert(tree, 4);
73     insert(tree, 8);
74     insert(tree, 3);
75     insert(tree, 2);
76     insert(tree, 6);
77     insert(tree, 10);
78     insert(tree, 3);
79     insert(tree, 5);
80     insert(tree, 4);
81     insert(tree, 8);
82
83     puts("Pre order: ");
84 }
85
86 int main() {
87     /* Delete this text */
88     t_node* tree = new_node(5);
89     insert(tree, 3);
90     insert(tree, 5);
91     insert(tree, 4);
92     insert(tree, 8);
93     insert(tree, 3);
94     insert(tree, 2);
95     insert(tree, 6);
96     insert(tree, 10);
97     insert(tree, 3);
98     insert(tree, 5);
99     insert(tree, 4);
100    insert(tree, 8);
101
102    puts("Pre order: ");
103 }
104
105 int main() {
106     /* Delete this text */
107     t_node* tree = new_node(5);
108     insert(tree, 3);
109     insert(tree, 5);
110     insert(tree, 4);
111     insert(tree, 8);
112     insert(tree, 3);
113     insert(tree, 2);
114     insert(tree, 6);
115     insert(tree, 10);
116     insert(tree, 3);
117     insert(tree, 5);
118     insert(tree, 4);
119     insert(tree, 8);
120
121     puts("Pre order: ");
122 }
123
124 int main() {
125     /* Delete this text */
126     t_node* tree = new_node(5);
127     insert(tree, 3);
128     insert(tree, 5);
129     insert(tree, 4);
130     insert(tree, 8);
131     insert(tree, 3);
132     insert(tree, 2);
133     insert(tree, 6);
134     insert(tree, 10);
135     insert(tree, 3);
136     insert(tree, 5);
137     insert(tree, 4);
138     insert(tree, 8);
139
140     puts("Pre order: ");
141 }
142
143 int main() {
144     /* Delete this text */
145     t_node* tree = new_node(5);
146     insert(tree, 3);
147     insert(tree, 5);
148     insert(tree, 4);
149     insert(tree, 8);
150     insert(tree, 3);
151     insert(tree, 2);
152     insert(tree, 6);
153     insert(tree, 10);
154     insert(tree, 3);
155     insert(tree, 5);
156     insert(tree, 4);
157     insert(tree, 8);
158
159     puts("Pre order: ");
160 }
161
162 int main() {
163     /* Delete this text */
164     t_node* tree = new_node(5);
165     insert(tree, 3);
166     insert(tree, 5);
167     insert(tree, 4);
168     insert(tree, 8);
169     insert(tree, 3);
170     insert(tree, 2);
171     insert(tree, 6);
172     insert(tree, 10);
173     insert(tree, 3);
174     insert(tree, 5);
175     insert(tree, 4);
176     insert(tree, 8);
177
178     puts("Pre order: ");
179 }
180
181 int main() {
182     /* Delete this text */
183     t_node* tree = new_node(5);
184     insert(tree, 3);
185     insert(tree, 5);
186     insert(tree, 4);
187     insert(tree, 8);
188     insert(tree, 3);
189     insert(tree, 2);
190     insert(tree, 6);
191     insert(tree, 10);
192     insert(tree, 3);
193     insert(tree, 5);
194     insert(tree, 4);
195     insert(tree, 8);
196
197     puts("Pre order: ");
198 }
199
200 int main() {
201     /* Delete this text */
202     t_node* tree = new_node(5);
203     insert(tree, 3);
204     insert(tree, 5);
205     insert(tree, 4);
206     insert(tree, 8);
207     insert(tree, 3);
208     insert(tree, 2);
209     insert(tree, 6);
210     insert(tree, 10);
211     insert(tree, 3);
212     insert(tree, 5);
213     insert(tree, 4);
214     insert(tree, 8);
215
216     puts("Pre order: ");
217 }
218
219 int main() {
220     /* Delete this text */
221     t_node* tree = new_node(5);
222     insert(tree, 3);
223     insert(tree, 5);
224     insert(tree, 4);
225     insert(tree, 8);
226     insert(tree, 3);
227     insert(tree, 2);
228     insert(tree, 6);
229     insert(tree, 10);
230     insert(tree, 3);
231     insert(tree, 5);
232     insert(tree, 4);
233     insert(tree, 8);
234
235     puts("Pre order: ");
236 }
237
238 int main() {
239     /* Delete this text */
240     t_node* tree = new_node(5);
241     insert(tree, 3);
242     insert(tree, 5);
243     insert(tree, 4);
244     insert(tree, 8);
245     insert(tree, 3);
246     insert(tree, 2);
247     insert(tree, 6);
248     insert(tree, 10);
249     insert(tree, 3);
250     insert(tree, 5);
251     insert(tree, 4);
252     insert(tree, 8);
253
254     puts("Pre order: ");
255 }
256
257 int main() {
258     /* Delete this text */
259     t_node* tree = new_node(5);
260     insert(tree, 3);
261     insert(tree, 5);
262     insert(tree, 4);
263     insert(tree, 8);
264     insert(tree, 3);
265     insert(tree, 2);
266     insert(tree, 6);
267     insert(tree, 10);
268     insert(tree, 3);
269     insert(tree, 5);
270     insert(tree, 4);
271     insert(tree, 8);
272
273     puts("Pre order: ");
274 }
275
276 int main() {
277     /* Delete this text */
278     t_node* tree = new_node(5);
279     insert(tree, 3);
280     insert(tree, 5);
281     insert(tree, 4);
282     insert(tree, 8);
283     insert(tree, 3);
284     insert(tree, 2);
285     insert(tree, 6);
286     insert(tree, 10);
287     insert(tree, 3);
288     insert(tree, 5);
289     insert(tree, 4);
290     insert(tree, 8);
291
292     puts("Pre order: ");
293 }
294
295 int main() {
296     /* Delete this text */
297     t_node* tree = new_node(5);
298     insert(tree, 3);
299     insert(tree, 5);
300     insert(tree, 4);
301     insert(tree, 8);
302     insert(tree, 3);
303     insert(tree, 2);
304     insert(tree, 6);
305     insert(tree, 10);
306     insert(tree, 3);
307     insert(tree, 5);
308     insert(tree, 4);
309     insert(tree, 8);
310
311     puts("Pre order: ");
312 }
313
314 int main() {
315     /* Delete this text */
316     t_node* tree = new_node(5);
317     insert(tree, 3);
318     insert(tree, 5);
319     insert(tree, 4);
320     insert(tree, 8);
321     insert(tree, 3);
322     insert(tree, 2);
323     insert(tree, 6);
324     insert(tree, 10);
325     insert(tree, 3);
326     insert(tree, 5);
327     insert(tree, 4);
328     insert(tree, 8);
329
330     puts("Pre order: ");
331 }
332
333 int main() {
334     /* Delete this text */
335     t_node* tree = new_node(5);
336     insert(tree, 3);
337     insert(tree, 5);
338     insert(tree, 4);
339     insert(tree, 8);
340     insert(tree, 3);
341     insert(tree, 2);
342     insert(tree, 6);
343     insert(tree, 10);
344     insert(tree, 3);
345     insert(tree, 5);
346     insert(tree, 4);
347     insert(tree, 8);
348
349     puts("Pre order: ");
350 }
351
352 int main() {
353     /* Delete this text */
354     t_node* tree = new_node(5);
355     insert(tree, 3);
356     insert(tree, 5);
357     insert(tree, 4);
358     insert(tree, 8);
359     insert(tree, 3);
360     insert(tree, 2);
361     insert(tree, 6);
362     insert(tree, 10);
363     insert(tree, 3);
364     insert(tree, 5);
365     insert(tree, 4);
366     insert(tree, 8);
367
368     puts("Pre order: ");
369 }
370
371 int main() {
372     /* Delete this text */
373     t_node* tree = new_node(5);
374     insert(tree, 3);
375     insert(tree, 5);
376     insert(tree, 4);
377     insert(tree, 8);
378     insert(tree, 3);
379     insert(tree, 2);
380     insert(tree, 6);
381     insert(tree, 10);
382     insert(tree, 3);
383     insert(tree, 5);
384     insert(tree, 4);
385     insert(tree, 8);
386
387     puts("Pre order: ");
388 }
389
390 int main() {
391     /* Delete this text */
392     t_node* tree = new_node(5);
393     insert(tree, 3);
394     insert(tree, 5);
395     insert(tree, 4);
396     insert(tree, 8);
397     insert(tree, 3);
398     insert(tree, 2);
399     insert(tree, 6);
400     insert(tree, 10);
401     insert(tree, 3);
402     insert(tree, 5);
403     insert(tree, 4);
404     insert(tree, 8);
405
406     puts("Pre order: ");
407 }
408
409 int main() {
410     /* Delete this text */
411     t_node* tree = new_node(5);
412     insert(tree, 3);
413     insert(tree, 5);
414     insert(tree, 4);
415     insert(tree, 8);
416     insert(tree, 3);
417     insert(tree, 2);
418     insert(tree, 6);
419     insert(tree, 10);
420     insert(tree, 3);
421     insert(tree, 5);
422     insert(tree, 4);
423     insert(tree, 8);
424
425     puts("Pre order: ");
426 }
427
428 int main() {
429     /* Delete this text */
430     t_node* tree = new_node(5);
431     insert(tree, 3);
432     insert(tree, 5);
433     insert(tree, 4);
434     insert(tree, 8);
435     insert(tree, 3);
436     insert(tree, 2);
437     insert(tree, 6);
438     insert(tree, 10);
439     insert(tree, 3);
440     insert(tree, 5);
441     insert(tree, 4);
442     insert(tree, 8);
443
444     puts("Pre order: ");
445 }
446
447 int main() {
448     /* Delete this text */
449     t_node* tree = new_node(5);
450     insert(tree, 3);
451     insert(tree, 5);
452     insert(tree, 4);
453     insert(tree, 8);
454     insert(tree, 3);
455     insert(tree, 2);
456     insert(tree, 6);
457     insert(tree, 10);
458     insert(tree, 3);
459     insert(tree, 5);
460     insert(tree, 4);
461     insert(tree, 8);
462
463     puts("Pre order: ");
464 }
465
466 int main() {
467     /* Delete this text */
468     t_node* tree = new_node(5);
469     insert(tree, 3);
470     insert(tree, 5);
471     insert(tree, 4);
472     insert(tree, 8);
473     insert(tree, 3);
474     insert(tree, 2);
475     insert(tree, 6);
476     insert(tree, 10);
477     insert(tree, 3);
478     insert(tree, 5);
479     insert(tree, 4);
480     insert(tree, 8);
481
482     puts("Pre order: ");
483 }
484
485 int main() {
486     /* Delete this text */
487     t_node* tree = new_node(5);
488     insert(tree, 3);
489     insert(tree, 5);
490     insert(tree, 4);
491     insert(tree, 8);
492     insert(tree, 3);
493     insert(tree, 2);
494     insert(tree, 6);
495     insert(tree, 10);
496     insert(tree, 3);
497     insert(tree, 5);
498     insert(tree, 4);
499     insert(tree, 8);
500
501     puts("Pre order: ");
502 }
503
504 int main() {
505     /* Delete this text */
506     t_node* tree = new_node(5);
507     insert(tree, 3);
508     insert(tree, 5);
509     insert(tree, 4);
510     insert(tree, 8);
511     insert(tree, 3);
512     insert(tree, 2);
513     insert(tree, 6);
514     insert(tree, 10);
515     insert(tree, 3);
516     insert(tree, 5);
517     insert(tree, 4);
518     insert(tree, 8);
519
520     puts("Pre order: ");
521 }
522
523 int main() {
524     /* Delete this text */
525     t_node* tree = new_node(5);
526     insert(tree, 3);
527     insert(tree, 5);
528     insert(tree, 4);
529     insert(tree, 8);
530     insert(tree, 3);
531     insert(tree, 2);
532     insert(tree, 6);
533     insert(tree, 10);
534     insert(tree, 3);
535     insert(tree, 5);
536     insert(tree, 4);
537     insert(tree, 8);
538
539     puts("Pre order: ");
540 }
541
542 int main() {
543     /* Delete this text */
544     t_node* tree = new_node(5);
545     insert(tree, 3);
546     insert(tree, 5);
547     insert(tree, 4);
548     insert(tree, 8);
549     insert(tree, 3);
550     insert(tree, 2);
551     insert(tree, 6);
552     insert(tree, 10);
553     insert(tree, 3);
554     insert(tree, 5);
555     insert(tree, 4);
556     insert(tree, 8);
557
558     puts("Pre order: ");
559 }
560
561 int main() {
562     /* Delete this text */
563     t_node* tree = new_node(5);
564     insert(tree, 3);
565     insert(tree, 5);
566     insert(tree, 4);
567     insert(tree, 8);
568     insert(tree, 3);
569     insert(tree, 2);
570     insert(tree, 6);
571     insert(tree, 10);
572     insert(tree, 3);
573     insert(tree, 5);
574     insert(tree, 4);
575     insert(tree, 8);
576
577     puts("Pre order: ");
578 }
579
580 int main() {
581     /* Delete this text */
582     t_node* tree = new_node(5);
583     insert(tree, 3);
584     insert(tree, 5);
585     insert(tree, 4);
586     insert(tree, 8);
587     insert(tree, 3);
588     insert(tree, 2);
589     insert(tree, 6);
590     insert(tree, 10);
591     insert(tree, 3);
592     insert(tree, 5);
593     insert(tree, 4);
594     insert(tree, 8);
595
596     puts("Pre order: ");
597 }
598
599 int main() {
600     /* Delete this text */
601     t_node* tree = new_node(5);
602     insert(tree, 3);
603     insert(tree, 5);
604     insert(tree, 4);
605     insert(tree, 8);
606     insert(tree, 3);
607     insert(tree, 2);
608     insert(tree, 6);
609     insert(tree, 10);
610     insert(tree, 3);
611     insert(tree, 5);
612     insert(tree, 4);
613     insert(tree, 8);
614
615     puts("Pre order: ");
616 }
617
618 int main() {
619     /* Delete this text */
620     t_node* tree = new_node(5);
621     insert(tree, 3);
622     insert(tree, 5);
623     insert(tree, 4);
624     insert(tree, 8);
625     insert(tree, 3);
626     insert(tree, 2);
627     insert(tree, 6);
628     insert(tree, 10);
629     insert(tree, 3);
630     insert(tree, 5);
631     insert(tree, 4);
632     insert(tree, 8);
633
634     puts("Pre order: ");
635 }
636
637 int main() {
638     /* Delete this text */
639     t_node* tree = new_node(5);
640     insert(tree, 3);
641     insert(tree, 5);
642     insert(tree, 4);
643     insert(tree, 8);
644     insert(tree, 3);
645     insert(tree, 2);
646     insert(tree, 6);
647     insert(tree, 10);
648     insert(tree, 3);
649     insert(tree, 5);
650     insert(tree, 4);
651     insert(tree, 8);
652
653     puts("Pre order: ");
654 }
655
656 int main() {
657     /* Delete this text */
658     t_node* tree = new_node(5);
659     insert(tree, 3);
660     insert(tree, 5);
661     insert(tree, 4);
662     insert(tree, 8);
663     insert(tree, 3);
664     insert(tree, 2);
665     insert(tree, 6);
666     insert(tree, 10);
667     insert(tree, 3);
668     insert(tree, 5);
669     insert(tree, 4);
670     insert(tree, 8);
671
672     puts("Pre order: ");
673 }
674
675 int main() {
676     /* Delete this text */
677     t_node* tree = new_node(5);
678     insert(tree, 3);
679     insert(tree, 5);
680     insert(tree, 4);
681     insert(tree, 8);
682     insert(tree, 3);
683     insert(tree, 2);
684     insert(tree, 6);
685     insert(tree, 10);
686     insert(tree, 3);
687     insert(tree, 5);
688     insert(tree, 4);
689     insert(tree, 8);
690
691     puts("Pre order: ");
692 }
693
694 int main() {
695     /* Delete this text */
696     t_node* tree = new_node(5);
697     insert(tree, 3);
698     insert(tree, 5);
699     insert(tree, 4);
700     insert(tree, 8);
701     insert(tree, 3);
702     insert(tree, 2);
703     insert(tree, 6);
704     insert(tree, 10);
705     insert(tree, 3);
706     insert(tree, 5);
707     insert(tree, 4);
708     insert(tree, 8);
709
710     puts("Pre order: ");
711 }
712
713 int main() {
714     /* Delete this text */
715     t_node* tree = new_node(5);
716     insert(tree, 3);
717     insert(tree, 5);
718     insert(tree, 4);
719     insert(tree, 8);
720     insert(tree, 3);
721     insert(tree, 2);
722     insert(tree, 6);
723     insert(tree, 10);
724     insert(tree, 3);
725     insert(tree, 5);
726     insert(tree, 4);
727     insert(tree, 8);
728
729     puts("Pre order: ");
730 }
731
732 int main() {
733     /* Delete this text */
734     t_node* tree = new_node(5);
735     insert(tree, 3);
736     insert(tree, 5);
737     insert(tree, 4);
738     insert(tree, 8);
739     insert(tree, 3);
740     insert(tree, 2);
741     insert(tree, 6);
742     insert(tree, 10);
743     insert(tree, 3);
744     insert(tree, 5);
745     insert(tree, 4);
746     insert(tree, 8);
747
748     puts("Pre order: ");
749 }
750
751 int main() {
752     /* Delete this text */
753     t_node* tree = new_node(5);
754     insert(tree, 3);
755     insert(tree, 5);
756     insert(tree, 4);
757     insert(tree, 8);
758     insert(tree, 3);
759     insert(tree, 2);
760     insert(tree, 6);
761     insert(tree, 10);
762     insert(tree, 3);
763     insert(tree, 5);
764     insert(tree, 4);
765     insert(tree, 8);
766
767     puts("Pre order: ");
768 }
769
770 int main() {
771     /* Delete this text */
772     t_node* tree = new_node(5);
773     insert(tree, 3);
774     insert(tree, 5);
775     insert(tree, 4);
776     insert(tree, 8);
777     insert(tree, 3);
778     insert(tree, 2);
779     insert(tree, 6);
780     insert(tree, 10);
781     insert(tree, 3);
782     insert(tree, 5);
783     insert(tree, 4);
784     insert(tree, 8);
785
786     puts("Pre order: ");
787 }
788
789 int main() {
790     /* Delete this text */
791     t_node* tree = new_node(5);
792     insert(tree, 3);
793     insert(tree, 5);
794     insert(tree, 4);
795     insert(tree, 8);
796     insert(tree, 3);
797     insert(tree, 2);
798     insert(tree, 6);
799     insert(tree, 10);
800     insert(tree, 3);
801     insert(tree, 5);
802     insert(tree, 4);
803     insert(tree, 8);
804
805     puts("Pre order: ");
806 }
807
808 int main() {
809     /* Delete this text */
810     t_node* tree = new_node(5);
811     insert(tree, 3);
812     insert(tree, 5);
813     insert(tree, 4);
814     insert(tree, 8);
815     insert(tree, 3);
816     insert(tree, 2);
817     insert(tree, 6);
818     insert(tree, 10);
819     insert(tree, 3);
820     insert(tree, 5);
821     insert(tree, 4);
822     insert(tree, 8);
823
824     puts("Pre order: ");
825 }
826
827 int main() {
828     /* Delete this text */
829     t_node* tree = new_node(5);
830     insert(tree, 3);
831     insert(tree, 5);
832     insert(tree, 4);
833     insert(tree, 8);
834     insert(tree, 3);
835     insert(tree, 2);
836     insert(tree, 6);
837     insert(tree, 10);
838     insert(tree, 3);
839     insert(tree, 5);
840     insert(tree, 4);
841     insert(tree, 8);
842
843     puts("Pre order: ");
844 }
845
846 int main() {
847     /* Delete this text */
848     t_node* tree = new_node(5);
849     insert(tree, 3);
850     insert(tree, 5);
851     insert(tree, 4);
852     insert(tree, 8);
853     insert(tree, 3);
854     insert(tree, 2);
855     insert(tree, 6);
856     insert(tree, 10);
857     insert(tree, 3);
858     insert(tree, 5);
859     insert(tree, 4);
860     insert(tree, 8);
861
862     puts("Pre order: ");
863 }
864
865 int main() {
866     /* Delete this text */
867     t_node* tree = new_node(5);
868     insert(tree, 3);
869     insert(tree, 5);
870     insert(tree, 4);
871     insert(tree, 8);
872     insert(tree, 3);
873     insert(tree, 2);
874     insert(tree, 6);
875     insert(tree, 10);
876     insert(tree, 3);
877     insert(tree, 5);
878     insert(tree, 4);
879     insert(tree, 8);
880
881     puts("Pre order: ");
882 }
883
884 int main() {
885     /* Delete this text */
886     t_node* tree = new_node(5);
887     insert(tree, 3);
888     insert(tree, 5);
889     insert(tree, 4);
890     insert(tree, 8);
891     insert(tree, 3);
892     insert(tree, 2);
893     insert(tree, 6);
894     insert(tree, 10);
895     insert(tree, 3);
896     insert(tree, 5);
897     insert(tree, 4);
898     insert(tree, 8);
899
900     puts("Pre order: ");
901 }
902
903 int main() {
904     /* Delete this text */
905     t_node* tree = new_node(5);
906     insert(tree, 3);
907     insert(tree, 5);
908     insert(tree, 4);
909     insert(tree, 8);
910     insert(tree, 3);
911     insert(tree, 2);
912     insert(tree, 6);
913     insert(tree, 10);
914     insert(tree, 3);
915     insert(tree, 5);
916     insert(tree, 4);
917     insert(tree, 8);
918
919     puts("Pre order: ");
920 }
921
922 int main() {
923     /* Delete this text */
924     t_node* tree = new_node(5);
925     insert(tree, 3);
926     insert(tree, 5);
927     insert(tree, 4);
928     insert(tree, 8);
929     insert(tree, 3);
930     insert(tree, 2);
931     insert(tree, 6);
932     insert(tree, 10);
933     insert(tree, 3);
934     insert(tree, 5);
935     insert(tree, 4);
936     insert(tree, 8);
937
938     puts("Pre order: ");
939 }
940
941 int main() {
942     /* Delete this text */
943     t_node* tree = new_node(5);
944     insert(tree, 3);
945     insert(tree, 5);
946     insert(tree, 4);
947     insert(tree, 8);
948     insert(tree, 3);
949     insert(tree, 2);
950     insert(tree, 6);
951     insert(tree, 10);
952     insert(tree, 3);
953     insert(tree, 5);
954     insert(tree, 4);
955     insert(tree, 8);
956
957     puts("Pre order: ");
958 }
959
960 int main() {
961     /* Delete this text */
962     t_node* tree = new_node(5);
963     insert(tree, 3);
964     insert(tree, 5);
965     insert(tree, 4);
966     insert(tree, 8);
967     insert(tree, 3);
968     insert(tree, 2);
969     insert(tree, 6);
970     insert(tree, 10);
971     insert(tree, 3);
972     insert(tree, 5);
973     insert(tree, 4);
974     insert(tree, 8);
975
976     puts("Pre order: ");
977 }
978
979 int main() {
980     /* Delete this text */
981     t_node* tree = new_node(5);
982     insert(tree, 3);
983     insert(tree, 5);
984     insert(tree, 4);
985     insert(tree, 8);
986     insert(tree, 3);
987     insert(tree, 2);
988     insert(tree, 6);
989     insert(tree, 10);
990     insert(tree, 3);
991     insert(tree, 5);
992     insert(tree, 4);
993     insert(tree, 8);
994
995     puts("Pre order: ");
996 }
997
998 int main() {
999     /* Delete this text */
1000    t_node* tree = new_node(5);
1001    insert(tree, 3);
1002    insert(tree, 5);
1003    insert(tree, 4);
1004    insert(tree, 8);
1005    insert(tree, 3);
1006    insert(tree, 2);
1007    insert(tree, 6);
1008    insert(tree, 10);
1009    insert(tree, 3);
1010    insert(tree, 5);
1011    insert(tree, 4);
1012    insert(tree, 8);
1013
1014    puts("Pre order: ");
1015 }
1016
1017 int main() {
1018     /* Delete this text */
1019     t_node* tree = new_node(5);
1020     insert(tree, 3);
1021     insert(tree, 5);
1022     insert(tree, 4);
1023     insert(tree, 8);
1024     insert(tree, 3);
1025     insert(tree, 2);
1026     insert(tree, 6);
1027     insert(tree, 10);
1028     insert(tree, 3);
1029     insert(tree, 5);
1030     insert(tree, 4);
1031     insert(tree, 8);
1032
1033     puts("Pre order: ");
1034 }
1035
1036 int main() {
1037     /* Delete this text */
1038     t_node* tree = new_node(5);
1039     insert(tree, 3);
1040     insert(tree, 5);
1041     insert(tree, 4);
1042     insert(tree, 8);
1043     insert(tree, 3);
1044     insert(tree, 2);
1045     insert(tree, 6);
1046     insert(tree, 10);
1047     insert(tree, 3);
1048     insert(tree, 5);
1049     insert(tree, 4);
1050     insert(tree, 8);
1051
1052     puts("Pre order: ");
1053 }
1054
1055 int main() {
1056     /* Delete this text */
1057     t_node* tree = new_node(5);
1058     insert(tree, 3);
1059     insert(tree, 5);
1060     insert(tree, 4);
1061     insert(tree, 8);
1062     insert(tree, 3);
1063     insert(tree, 2);
1064     insert(tree, 6);
1065     insert(tree, 10);
1066     insert(tree, 3);
1067     insert(tree, 5);
1068     insert(tree, 4);
1069     insert(tree, 8);
1070
1071     puts("Pre order: ");
1072 }
1073
1074 int main() {
1075     /* Delete this text */
1076     t_node* tree = new_node(5);
1077     insert(tree, 3);
1078     insert(tree, 5);
1079     insert(tree, 4);
1080     insert(tree, 8);
1081     insert(tree, 3);
1082     insert(tree, 2);
1083     insert(tree, 6);
1084     insert(tree, 10);
1085     insert(tree, 3);
1086     insert(tree, 5);
1087     insert(tree, 4);
1088     insert(tree, 8);
1089
1090     puts("Pre order: ");
1091 }
1092
1093 int main() {
1094     /* Delete this text */
1095     t_node* tree = new_node(5);
1096     insert(tree, 3);
1097     insert(tree, 5);
1098     insert(tree, 4);
1099     insert(tree, 8);
1100     insert(tree, 3);
1101     insert(tree, 2);
1102     insert(tree, 6);
1103     insert(tree, 10);
1104     insert(tree, 3);
1105     insert(tree, 5);
1106     insert(tree, 4);
1107     insert(tree, 8);
1108
1109     puts("Pre order: ");
1110 }
1111
1112 int main() {
1113     /* Delete this text */
1114     t_node* tree = new_node(5);
1115     insert(tree, 3);
1116     insert(tree, 5);
1117     insert(tree, 4);
1118     insert(tree, 8);
1119     insert(tree, 3);
1120     insert(tree, 2);
1121     insert(tree, 6);
1122     insert(tree, 10);
1123     insert(tree, 3);
1124     insert(tree, 5);
1125     insert(tree, 4);
1126     insert(tree, 8);
1127
1128     puts("Pre order: ");
1129 }
1130
1131 int main() {
1132     /* Delete this text */
1133     t_node* tree = new_node(5);
1134     insert(tree, 3);
1135     insert(tree, 5);
1136     insert(tree, 4);
1137     insert(tree, 8);
1138     insert(tree, 3);
1139     insert(tree, 2);
1140     insert(tree, 6);
1141     insert(tree, 10);
1142     insert(tree, 3);
1143     insert(tree, 5);
1144     insert(tree, 4);
1145     insert(tree, 8);
1146
1147     puts("Pre order: ");
1148 }
1149
1150 int main() {
115
```

14. The `t_` in `t_node` is redundant: change all occurrences of `t_node` with `node` by `:%s/t_node/node/g`. Here `s` is short for substitute and has the following usage: `[range]s/pattern/string/[flags] [count]`. `%` is a range which is short for `1,$` meaning from the first to last line. In general ranges have the form `beginning,end`. `g` is a flag which replaces every occurrence on each line being replaced, without it only the first occurrence is replaced.



```
ari@administrator-hp-dv4: ~/CompSci/Computer-Architecture/recitation1
59
60 void free_tree(node* root) {
61     if (root == NULL)
62         return;
63     free_tree(root->left);
64     free_tree(root->right);
65     free(root);
66 }
67
68 int main() {
69     /* Delete this text */
70     node* tree = new_node(5);
71     insert(tree, 3);
72     insert(tree, 5);
73     insert(tree, 4);
74     insert(tree, 8);
75     insert(tree, 3);
76     insert(tree, 2);
77     insert(tree, 6);
78     insert(tree, 10);
79
80     puts("Pre order: ");
81     print_pre_order(tree);
82 }
13 substitutions on 12 lines 70,5 87%
```

15. Never mind, the `t_` is actually useful: undo the previous change with `u`. But you can always redo a change with the command `:redo`.
16. Exit vim. Then enter `vimtutor`, and navigate around a bit. You don't need to follow the instructions but make sure the students are aware of its existence.

Important notes:

1. It is not necessary to follow these instructions verbatim, these are only some guidelines I think may be useful. Alternatively you could just follow vimtutor entirely.
2. To exit multiple vim windows at once (for example an editor and a terminal window), do `:qa`. To also save, do `:wqa`. `:q` closes only the current selected window, you must add the `a` to close all windows.
3. To scroll within vim's terminal, do `CTRL-w N`. This brings you to "Terminal Normal mode", where you can navigate around the terminal similar to as you would in normal mode.
4. If you see that vim says something like "recording @q" at the bottom — you have accidentally started recording a key. Just hit `q` in normal mode to stop this. It's not significant but it can be annoying and a student may ask you what it means.
5. If you're typing keys but they're not appearing: you have most likely accidentally pressed `CTRL-s` and froze your terminal. Hit `CTRL-q` to unfreeze.
6. You can always do `:help <command>` to get help on a command.