

What's New? Program.cs

LLPARSER LL1Parser tokens

```
1 using System;
2 using System.Collections.Generic;
3
4 class LL1Parser
5 {
6     private string[] tokens;
7     private int index = 0;
8
9     1 reference
10    public LL1Parser(string input)
11    {
12        tokens = input.Split(' ', StringSplitOptions.RemoveEmptyEntries);
13    }
14
15    11 references
16    private string CurrentToken => index < tokens.Length ? tokens[index] : "$"; // End of input ma
17
18    9 references
19    private void Match(string expectedToken)
20    {
21        if (CurrentToken == expectedToken)
22        {
23            index++;
24        }
25    }
26
27    1 reference
28    public void Parse()
29    {
30        S();
31        if (CurrentToken != "$")
32        {
33            throw new Exception("Syntax Error: Extra input remaining");
34        }
35        Console.WriteLine("Parsing Successful!");
36    }
37
38    1 reference
39    private void S()
40    {
41        if (CurrentToken == "e")
42        {
43            Match("e");
44        }
45    }
46
47    private void S1()
48    {
49        Match("(");
50        S();
51        Match(")");
52    }
53
54    private void S2()
55    {
56        Match("(");
57        S1();
58        Match(")");
59    }
60
61    private void S3()
62    {
63        Match("(");
64        S2();
65        Match(")");
66    }
67
68    private void S4()
69    {
70        Match("(");
71        S3();
72        Match(")");
73    }
74
75    private void S5()
76    {
77        Match("(");
78        S4();
79        Match(")");
80    }
81
82    private void S6()
83    {
84        Match("(");
85        S5();
86        Match(")");
87    }
88
89    private void S7()
90    {
91        Match("(");
92        S6();
93        Match(")");
94    }
95
96    private void S8()
97    {
98        Match("(");
99        S7();
100       Match(")");
101    }
102
103    private void S9()
104    {
105        Match("(");
106        S8();
107        Match(")");
108    }
109
110    private void S10()
111    {
112        Match("(");
113        S9();
114        Match(")");
115    }
116
117    private void S11()
118    {
119        Match("(");
120        S10();
121        Match(")");
122    }
123
124    private void S12()
125    {
126        Match("(");
127        S11();
128        Match(")");
129    }
130
131    private void S13()
132    {
133        Match("(");
134        S12();
135        Match(")");
136    }
137
138    private void S14()
139    {
140        Match("(");
141        S13();
142        Match(")");
143    }
144
145    private void S15()
146    {
147        Match("(");
148        S14();
149        Match(")");
150    }
151
152    private void S16()
153    {
154        Match("(");
155        S15();
156        Match(")");
157    }
158
159    private void S17()
160    {
161        Match("(");
162        S16();
163        Match(")");
164    }
165
166    private void S18()
167    {
168        Match("(");
169        S17();
170        Match(")");
171    }
172
173    private void S19()
174    {
175        Match("(");
176        S18();
177        Match(")");
178    }
179
180    private void S20()
181    {
182        Match("(");
183        S19();
184        Match(")");
185    }
186
187    private void S21()
188    {
189        Match("(");
190        S20();
191        Match(")");
192    }
193
194    private void S22()
195    {
196        Match("(");
197        S21();
198        Match(")");
199    }
200
201    private void S23()
202    {
203        Match("(");
204        S22();
205        Match(")");
206    }
207
208    private void S24()
209    {
210        Match("(");
211        S23();
212        Match(")");
213    }
214
215    private void S25()
216    {
217        Match("(");
218        S24();
219        Match(")");
220    }
221
222    private void S26()
223    {
224        Match("(");
225        S25();
226        Match(")");
227    }
228
229    private void S27()
230    {
231        Match("(");
232        S26();
233        Match(")");
234    }
235
236    private void S28()
237    {
238        Match("(");
239        S27();
240        Match(")");
241    }
242
243    private void S29()
244    {
245        Match("(");
246        S28();
247        Match(")");
248    }
249
250    private void S30()
251    {
252        Match("(");
253        S29();
254        Match(")");
255    }
256
257    private void S31()
258    {
259        Match("(");
260        S30();
261        Match(")");
262    }
263
264    private void S32()
265    {
266        Match("(");
267        S31();
268        Match(")");
269    }
270
271    private void S33()
272    {
273        Match("(");
274        S32();
275        Match(")");
276    }
277
278    private void S34()
279    {
280        Match("(");
281        S33();
282        Match(")");
283    }
284
285    private void S35()
286    {
287        Match("(");
288        S34();
289        Match(")");
290    }
291
292    private void S36()
293    {
294        Match("(");
295        S35();
296        Match(")");
297    }
298
299    private void S37()
300    {
301        Match("(");
302        S36();
303        Match(")");
304    }
305
306    private void S38()
307    {
308        Match("(");
309        S37();
310        Match(")");
311    }
312
313    private void S39()
314    {
315        Match("(");
316        S38();
317        Match(")");
318    }
319
320    private void S40()
321    {
322        Match("(");
323        S39();
324        Match(")");
325    }
326
327    private void S41()
328    {
329        Match("(");
330        S40();
331        Match(")");
332    }
333
334    private void S42()
335    {
336        Match("(");
337        S41();
338        Match(")");
339    }
340
341    private void S43()
342    {
343        Match("(");
344        S42();
345        Match(")");
346    }
347
348    private void S44()
349    {
350        Match("(");
351        S43();
352        Match(")");
353    }
354
355    private void S45()
356    {
357        Match("(");
358        S44();
359        Match(")");
360    }
361
362    private void S46()
363    {
364        Match("(");
365        S45();
366        Match(")");
367    }
368
369    private void S47()
370    {
371        Match("(");
372        S46();
373        Match(")");
374    }
375
376    private void S48()
377    {
378        Match("(");
379        S47();
380        Match(")");
381    }
382
383    private void S49()
384    {
385        Match("(");
386        S48();
387        Match(")");
388    }
389
390    private void S50()
391    {
392        Match("(");
393        S49();
394        Match(")");
395    }
396
397    private void S51()
398    {
399        Match("(");
400        S50();
401        Match(")");
402    }
403
404    private void S52()
405    {
406        Match("(");
407        S51();
408        Match(")");
409    }
410
411    private void S53()
412    {
413        Match("(");
414        S52();
415        Match(")");
416    }
417
418    private void S54()
419    {
420        Match("(");
421        S53();
422        Match(")");
423    }
424
425    private void S55()
426    {
427        Match("(");
428        S54();
429        Match(")");
430    }
431
432    private void S56()
433    {
434        Match("(");
435        S55();
436        Match(")");
437    }
438
439    private void S57()
440    {
441        Match("(");
442        S56();
443        Match(")");
444    }
445
446    private void S58()
447    {
448        Match("(");
449        S57();
450        Match(")");
451    }
452
453    private void S59()
454    {
455        Match("(");
456        S58();
457        Match(")");
458    }
459
460    private void S60()
461    {
462        Match("(");
463        S59();
464        Match(")");
465    }
466
467    private void S61()
468    {
469        Match("(");
470        S60();
471        Match(")");
472    }
473
474    private void S62()
475    {
476        Match("(");
477        S61();
478        Match(")");
479    }
480
481    private void S63()
482    {
483        Match("(");
484        S62();
485        Match(")");
486    }
487
488    private void S64()
489    {
490        Match("(");
491        S63();
492        Match(")");
493    }
494
495    private void S65()
496    {
497        Match("(");
498        S64();
499        Match(")");
500    }
501
502    private void S66()
503    {
504        Match("(");
505        S65();
506        Match(")");
507    }
508
509    private void S67()
510    {
511        Match("(");
512        S66();
513        Match(")");
514    }
515
516    private void S68()
517    {
518        Match("(");
519        S67();
520        Match(")");
521    }
522
523    private void S69()
524    {
525        Match("(");
526        S68();
527        Match(")");
528    }
529
530    private void S70()
531    {
532        Match("(");
533        S69();
534        Match(")");
535    }
536
537    private void S71()
538    {
539        Match("(");
540        S70();
541        Match(")");
542    }
543
544    private void S72()
545    {
546        Match("(");
547        S71();
548        Match(")");
549    }
550
551    private void S73()
552    {
553        Match("(");
554        S72();
555        Match(")");
556    }
557
558    private void S74()
559    {
560        Match("(");
561        S73();
562        Match(")");
563    }
564
565    private void S75()
566    {
567        Match("(");
568        S74();
569        Match(")");
570    }
571
572    private void S76()
573    {
574        Match("(");
575        S75();
576        Match(")");
577    }
578
579    private void S77()
580    {
581        Match("(");
582        S76();
583        Match(")");
584    }
585
586    private void S78()
587    {
588        Match("(");
589        S77();
590        Match(")");
591    }
592
593    private void S79()
594    {
595        Match("(");
596        S78();
597        Match(")");
598    }
599
600    private void S80()
601    {
602        Match("(");
603        S79();
604        Match(")");
605    }
606
607    private void S81()
608    {
609        Match("(");
610        S80();
611        Match(")");
612    }
613
614    private void S82()
615    {
616        Match("(");
617        S81();
618        Match(")");
619    }
620
621    private void S83()
622    {
623        Match("(");
624        S82();
625        Match(")");
626    }
627
628    private void S84()
629    {
630        Match("(");
631        S83();
632        Match(")");
633    }
634
635    private void S85()
636    {
637        Match("(");
638        S84();
639        Match(")");
640    }
641
642    private void S86()
643    {
644        Match("(");
645        S85();
646        Match(")");
647    }
648
649    private void S87()
650    {
651        Match("(");
652        S86();
653        Match(")");
654    }
655
656    private void S88()
657    {
658        Match("(");
659        S87();
660        Match(")");
661    }
662
663    private void S89()
664    {
665        Match("(");
666        S88();
667        Match(")");
668    }
669
670    private void S90()
671    {
672        Match("(");
673        S89();
674        Match(")");
675    }
676
677    private void S91()
678    {
679        Match("(");
680        S90();
681        Match(")");
682    }
683
684    private void S92()
685    {
686        Match("(");
687        S91();
688        Match(")");
689    }
690
691    private void S93()
692    {
693        Match("(");
694        S92();
695        Match(")");
696    }
697
698    private void S94()
699    {
700        Match("(");
701        S93();
702        Match(")");
703    }
704
705    private void S95()
706    {
707        Match("(");
708        S94();
709        Match(")");
710    }
711
712    private void S96()
713    {
714        Match("(");
715        S95();
716        Match(")");
717    }
718
719    private void S97()
720    {
721        Match("(");
722        S96();
723        Match(")");
724    }
725
726    private void S98()
727    {
728        Match("(");
729        S97();
730        Match(")");
731    }
732
733    private void S99()
734    {
735        Match("(");
736        S98();
737        Match(")");
738    }
739
740    private void S100()
741    {
742        Match("(");
743        S99();
744        Match(")");
745    }
746
747    private void S101()
748    {
749        Match("(");
750        S100();
751        Match(")");
752    }
753
754    private void S102()
755    {
756        Match("(");
757        S101();
758        Match(")");
759    }
760
761    private void S103()
762    {
763        Match("(");
764        S102();
765        Match(")");
766    }
767
768    private void S104()
769    {
770        Match("(");
771        S103();
772        Match(")");
773    }
774
775    private void S105()
776    {
777        Match("(");
778        S104();
779        Match(")");
780    }
781
782    private void S106()
783    {
784        Match("(");
785        S105();
786        Match(")");
787    }
788
789    private void S107()
790    {
791        Match("(");
792        S106();
793        Match(")");
794    }
795
796    private void S108()
797    {
798        Match("(");
799        S107();
800        Match(")");
801    }
802
803    private void S109()
804    {
805        Match("(");
806        S108();
807        Match(")");
808    }
809
810    private void S110()
811    {
812        Match("(");
813        S109();
814        Match(")");
815    }
816
817    private void S111()
818    {
819        Match("(");
820        S110();
821        Match(")");
822    }
823
824    private void S112()
825    {
826        Match("(");
827        S111();
828        Match(")");
829    }
830
831    private void S113()
832    {
833        Match("(");
834        S112();
835        Match(")");
836    }
837
838    private void S114()
839    {
840        Match("(");
841        S113();
842        Match(")");
843    }
844
845    private void S115()
846    {
847        Match("(");
848        S114();
849        Match(")");
850    }
851
852    private void S116()
853    {
854        Match("(");
855        S115();
856        Match(")");
857    }
858
859    private void S117()
860    {
861        Match("(");
862        S116();
863        Match(")");
864    }
865
866    private void S118()
867    {
868        Match("(");
869        S117();
870        Match(")");
871    }
872
873    private void S119()
874    {
875        Match("(");
876        S118();
877        Match(")");
878    }
879
880    private void S120()
881    {
882        Match("(");
883        S119();
884        Match(")");
885    }
886
887    private void S121()
888    {
889        Match("(");
890        S120();
891        Match(")");
892    }
893
894    private void S122()
895    {
896        Match("(");
897        S121();
898        Match(")");
899    }
900
901    private void S123()
902    {
903        Match("(");
904        S122();
905        Match(")");
906    }
907
908    private void S124()
909    {
910        Match("(");
911        S123();
912        Match(")");
913    }
914
915    private void S125()
916    {
917        Match("(");
918        S124();
919        Match(")");
920    }
921
922    private void S126()
923    {
924        Match("(");
925        S125();
926        Match(")");
927    }
928
929    private void S127()
930    {
931        Match("(");
932        S126();
933        Match(")");
934    }
935
936    private void S128()
937    {
938        Match("(");
939        S127();
940        Match(")");
941    }
942
943    private void S129()
944    {
945        Match("(");
946        S128();
947        Match(")");
948    }
949
950    private void S130()
951    {
952        Match("(");
953        S129();
954        Match(")");
955    }
956
957    private void S131()
958    {
959        Match("(");
960        S130();
961        Match(")");
962    }
963
964    private void S132()
965    {
966        Match("(");
967        S131();
968        Match(")");
969    }
970
971    private void S133()
972    {
973        Match("(");
974        S132();
975        Match(")");
976    }
977
978    private void S134()
979    {
980        Match("(");
981        S133();
982        Match(")");
983    }
984
985    private void S135()
986    {
987        Match("(");
988        S134();
989        Match(")");
990    }
991
992    private void S136()
993    {
994        Match("(");
995        S135();
996        Match(")");
997    }
998
999    private void S137()
1000    {
1001        Match("(");
1002        S136();
1003        Match(")");
1004    }
1005
1006    private void S138()
1007    {
1008        Match("(");
1009        S137();
1010        Match(")");
1011    }
1012
1013    private void S139()
1014    {
1015        Match("(");
1016        S138();
1017        Match(")");
1018    }
1019
1020    private void S140()
1021    {
1022        Match("(");
1023        S139();
1024        Match(")");
1025    }
1026
1027    private void S141()
1028    {
1029        Match("(");
1030        S140();
1031        Match(")");
1032    }
1033
1034    private void S142()
1035    {
1036        Match("(");
1037        S141();
1038        Match(")");
1039    }
1040
1041    private void S143()
1042    {
1043        Match("(");
1044        S142();
1045        Match(")");
1046    }
1047
1048    private void S144()
1049    {
1050        Match("(");
1051        S143();
1052        Match(")");
1053    }
1054
1055    private void S145()
1056    {
1057        Match("(");
1058        S144();
1059        Match(")");
1060    }
1061
1062    private void S146()
1063    {
1064        Match("(");
1065        S145();
1066        Match(")");
1067    }
1068
1069    private void S147()
1070    {
1071        Match("(");
1072        S146();
1073        Match(")");
1074    }
1075
1076    private void S148()
1077    {
1078        Match("(");
1079        S147();
1080        Match(")");
1081    }
1082
1083    private void S149()
1084    {
1085        Match("(");
1086        S148();
1087        Match(")");
1088    }
1089
1090    private void S150()
1091    {
1092        Match("(");
1093        S149();
1094        Match(")");
1095    }
1096
1097    private void S151()
1098    {
1099        Match("(");
1100        S150();
1101        Match(")");
1102    }
1103
1104    private void S152()
1105    {
1106        Match("(");
1107        S151();
1108        Match(")");
1109    }
1110
1111    private void S153()
1112    {
1113        Match("(");
1114        S152();
1115        Match(")");
1116    }
1117
1118    private void S154()
1119    {
1120        Match("(");
1121        S153();
1122        Match(")");
1123    }
1124
1125    private void S155()
1126    {
1127        Match("(");
1128        S154();
1129        Match(")");
1130    }
1131
1132    private void S156()
1133    {
1134        Match("(");
1135        S155();
1136        Match(")");
1137    }
1138
1139    private void S157()
1140    {
1141        Match("(");
1142        S156();
1143        Match(")");
1144    }
1145
1146    private void S158()
1147    {
1148        Match("(");
1149        S157();
1150        Match(")");
1151    }
1152
1153    private void S159()
1154    {
1155        Match("(");
1156        S158();
1157        Match(")");
1158    }
1159
1160    private void S160()
1161    {
1162        Match("(");
1163        S159();
1164        Match(")");
1165    }
1166
1167    private void S161()
1168    {
1169        Match("(");
1170        S160();
1171        Match(")");
1172    }
1173
1174    private void S162()
1175    {
1176        Match("(");
1177        S161();
1178        Match(")");
1179    }
1180
1181    private void S163()
1182    {
1183        Match("(");
1184        S162();
1185        Match(")");
1186    }
1187
1188    private void S164()
1189    {
1190        Match("(");
1191        S163();
1192        Match(")");
1193    }
1194
1195    private void S165()
1196    {
1197        Match("(");
1198        S164();
1199        Match(")");
1200    }
1201
1202    private void S166()
1203    {
1204        Match("(");
1205        S165();
1206        Match(")");
1207    }
1208
1209    private void S167()
1210    {
1211        Match("(");
1212        S166();
1213        Match(")");
1214    }
1215
1216    private void S168()
1217    {
1218        Match("(");
1219        S167();
1220        Match(")");
1221    }
1222
1223    private void S169()
1224    {
1225        Match("(");
1226        S168();
1227        Match(")");
1228    }
1229
1230    private void S170()
1231    {
1232        Match("(");
1233        S169();
1234        Match(")");
1235    }
1236
1237    private void S171()
1238    {
1239        Match("(");
1240        S170();
1241        Match(")");
1242    }
1243
1244    private void S172()
1245    {
1246        Match("(");
1247        S171();
1248        Match(")");
1249    }
1250
1251    private void S173()
1252    {
1253        Match("(");
1254        S172();
1255        Match(")");
1256    }
1257
1258    private void S174()
1259    {
1260        Match("(");
1261        S173();
1262        Match(")");
1263    }
1264
1265    private void S175()
1266    {
1267        Match("(");
1268        S174();
1269        Match(")");
1270    }
1271
1272    private void S176()
1273    {
1274        Match("(");
1275        S175();
1276        Match(")");
1277    }
1278
1279    private void S177()
1280    {
1281        Match("(");
1282        S176();
1283        Match(")");
1284    }
1285
1286    private void S178()
1287    {
1288        Match("(");
1289        S177();
1290        Match(")");
1291    }
1292
1293    private void S179()
1294    {
1295        Match("(");
1296        S178();
1297        Match(")");
1298    }
1299
1300    private void S180()
1301    {
1302        Match("(");
1303        S179();
1304        Match(")");
1305    }
1306
1307    private void S181()
1308    {
1309        Match("(");
1310        S180();
1311        Match(")");
1312    }
1313
1314    private void S182()
1315    {
1316        Match("(");
1317        S181();
1318        Match(")");
1319    }
1320
1321    private void S183()
1322    {
1323        Match("(");
1324        S182();
1325        Match(")");
1326    }
1327
1328    private void S184()
1329    {
1330        Match("(");
1331        S183();
1332        Match(")");
1333    }
1334
1335    private void S185()
1336    {
1337        Match("(");
1338        S184();
1339        Match(")");
1340    }
1341
1342    private void S186()
1343    {
1344        Match("(");
1345        S185();
1346        Match(")");
1347    }
1348
1349    private void S187()
1350    {
1351        Match("(");
1352        S186();
1353        Match(")");
1354    }
1355
1356    private void S188()
1357    {
1358        Match("(");
1359        S187();
1360        Match(")");
1361    }
1362
1363    private void S189()
1364    {
1365        Match("(");
1366        S188();
1367        Match(")");
1368    }
1369
1370    private void S190()
1371    {
1372        Match("(");
1373        S189();
1374        Match(")");
1375    }
1376
1377    private void S191()
1378    {
1379        Match("(");
1380        S190();
1381        Match(")");
1382    }
1383
1384    private void S192()
1385    {
1386        Match("(");
1387        S191();
1388        Match(")");
1389    }
1390
1391    private void S193()
1392    {
1393        Match("(");
1394        S192();
1395        Match(")");
1396    }
1397
1398    private void S194()
1399    {
1400        Match("(");
1401        S193();
1402        Match(")");
1403    }
1404
1405    private void S195()
1406    {
1407        Match("(");
1408        S194();
1409        Match(")");
1410    }
1411
1412    private void S196()
1413    {
1414        Match("(");
1415        S195();
1416        Match(")");
1417    }
1418
1419    private void S197()
1420    {
1421        Match("(");
1422        S196();
1423        Match(")");
1424    }
1425
1426    private void S198()
1427    {
1428        Match("(");
1429        S197();
1430        Match(")");
1431    }
1432
1433    private void S199()
1434    {
1435        Match("(");
1436        S198();
1437        Match(")");
1438    }
1439
1440    private void S200()
1441    {
1442        Match("(");
1443        S199();
1444        Match(")");
1445    }
1446
1447    private void S201()
1448    {
1449        Match("(");
1450        S200();
1451        Match(")");
1452    }
1453
1454    private void S202()
1455    {
1456        Match("(");
1457        S201();
1458        Match(")");
1459    }
1460
1461    private void S203()
1462    {
1463        Match("(");
1464        S202();
1465        Match(")");
1466    }
1467
1468    private void S204()
1469    {
1470        Match("(");
1471        S203();
1472        Match(")");
1473    }
1474
1475    private void S205()
1476    {
1477        Match("(");
1478        S204();
1479        Match(")");
1480    }
1481
1482    private void S206()
1483    {
1484        Match("(");
1485        S205();
1486        Match(")");
1487    }
1488
1489    private void S207()
1490    {
1491        Match("(");
1492        S206();
1493        Match(")");
1494    }
1495
1496    private void S208()
1497    {
1498        Match("(");
1499        S207();
1500        Match(")");
1501    }
1502
1503    private void S209()
1504    {
1505        Match("(");
1506        S208();
1507        Match(")");
1508    }
1509
1510    private void S210()
1511    {
1512        Match("(");
1513        S209();
1514        Match(")");
1515    }
1516
1517    private void S211()
1518    {
1519        Match("(");
1520        S210();
1521        Match(")");
1522    }
1523
1524    private void S212()
1525    {
1526        Match("(");
1527        S211();
1528        Match(")");
1529    }
1530
1531    private void S213()
1532    {
1533        Match("(");
1534        S212();
1535        Match(")");
1536    }
1537
1538    private void S214()
1539    {
1540        Match("(");
```

```
What's New? Program.cs LL1Parser tokens
41 {
42     C();
43     Match("x");
44     Match("y");
45     SPrime();
46 }
47 else if (CurrentToken == "d")
48 {
49     Match("d");
50     Match("y");
51     SPrime();
52 }
53 else if (CurrentToken == "b")
54 {
55     Match("b");
56     SPrime();
57 }
58 else
59 {
60     throw new Exception($"Syntax Error in S: Unexpected token '{CurrentToken}'");
61 }
62 }
63 }
```

4 references

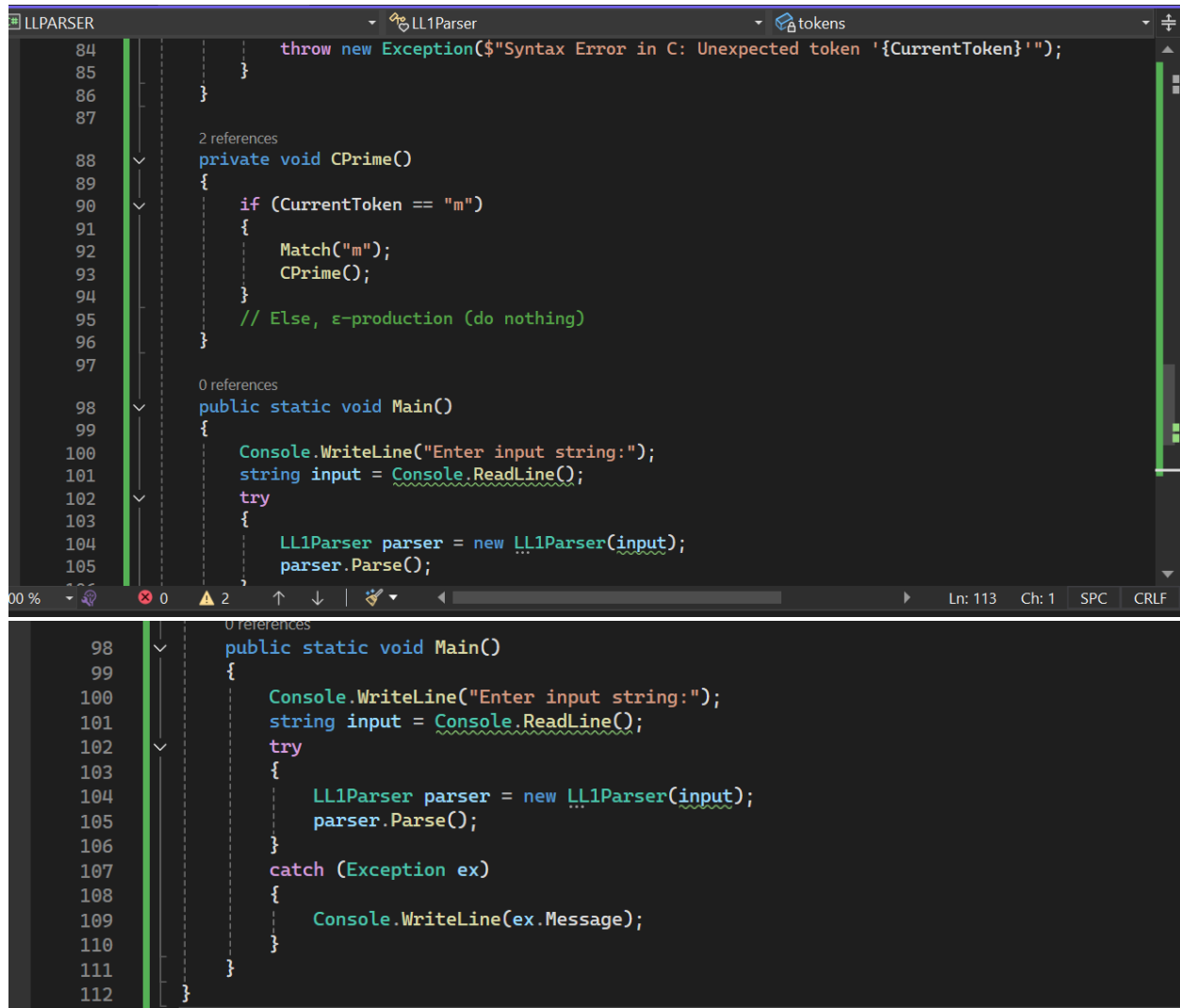
100 % 0 2 Ln: 113 Ch: 1 SPC CRLF

```
LL1Parser tokens
64 private void SPrime()
65 {
66     if (CurrentToken == "x")
67     {
68         Match("x");
69         Match("y");
70         SPrime();
71     }
72     // Else,  $\epsilon$ -production (do nothing)
73 }
74
75 private void C()
76 {
77     if (CurrentToken == "e")
78     {
79         Match("e");
80         CPrime();
81     }
82     else
83     {
84         throw new Exception($"Syntax Error in C: Unexpected token '{CurrentToken}'");
85     }
86 }
```

4 references

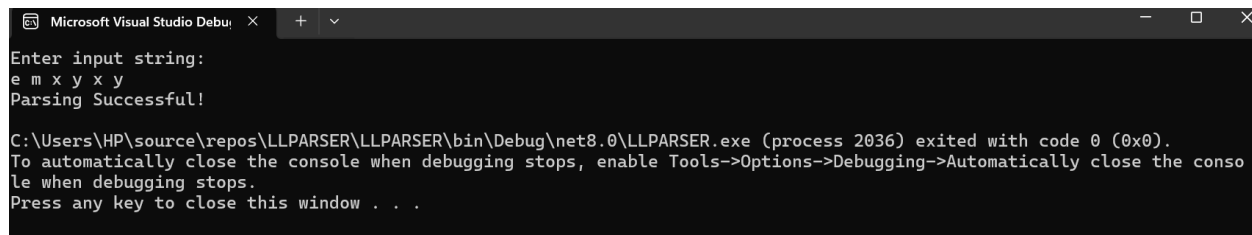
1 reference

100 % 0 2 Ln: 113 Ch: 1 SPC CRLF



```
84         throw new Exception($"Syntax Error in C: Unexpected token '{CurrentToken}");
85     }
86 }
87
88 2 references
89 private void CPrime()
90 {
91     if (CurrentToken == "m")
92     {
93         Match("m");
94         CPrime();
95     }
96     // Else, ε-production (do nothing)
97 }
98
99 0 references
100 public static void Main()
101 {
102     Console.WriteLine("Enter input string:");
103     string input = Console.ReadLine();
104     try
105     {
106         LL1Parser parser = new LL1Parser(input);
107         parser.Parse();
108     }
109 }
110
111 0 references
112 public static void Main()
113 {
114     Console.WriteLine("Enter input string:");
115     string input = Console.ReadLine();
116     try
117     {
118         LL1Parser parser = new LL1Parser(input);
119         parser.Parse();
120     }
121     catch (Exception ex)
122     {
123         Console.WriteLine(ex.Message);
124     }
125 }
```

OUTPUT:



```
Microsoft Visual Studio Debug:
Enter input string:
e m x y x y
Parsing Successful!

C:\Users\HP\source\repos\LLPARSER\LLPARSER\bin\Debug\net8.0\LLPARSER.exe (process 2036) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```