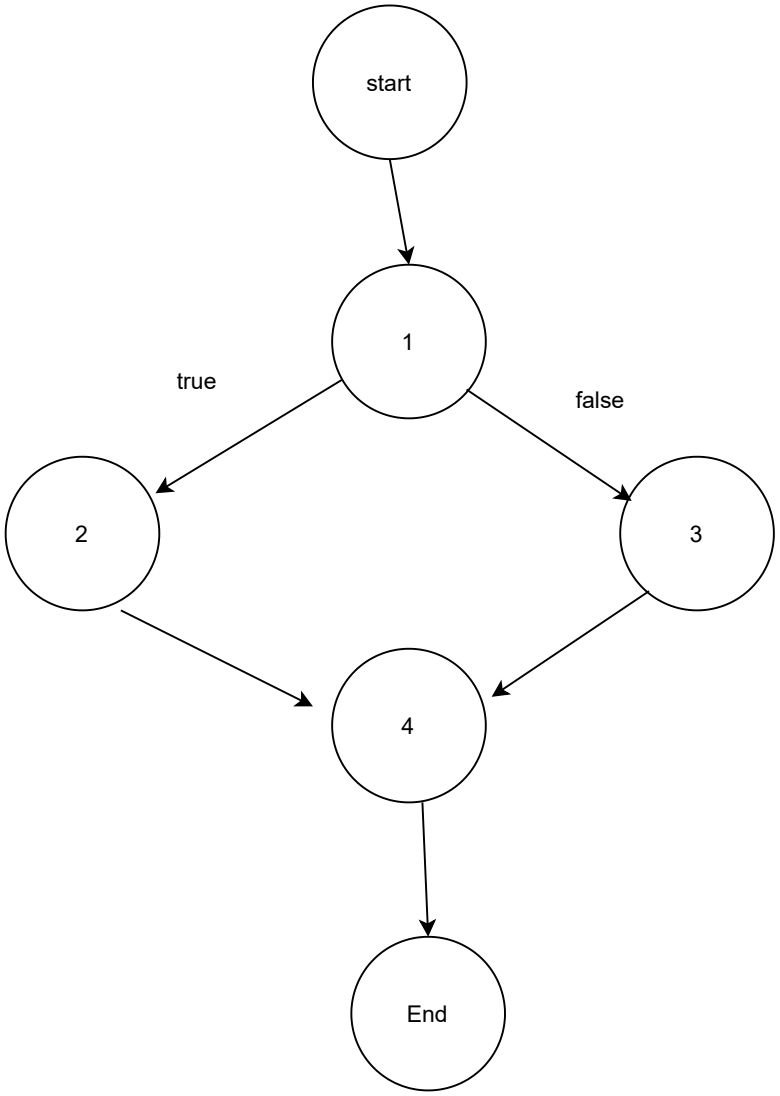
1. Open_character_stream

```
25    BufferedReader open_character_stream(String fname) {
26        BufferedReader br = null;
27        if (fname == null) {
28            br = new BufferedReader(new InputStreamReader(System.in));
29        } else {
30            try {
31                FileReader fr = new FileReader(fname);
32                br = new BufferedReader(fr);
33            } catch (FileNotFoundException e) {
34                System.out.print("The file " + fname +" doesn't exists\n");
35                e.printStackTrace();
36            }
37        }
38
39        return br;
40    }
```

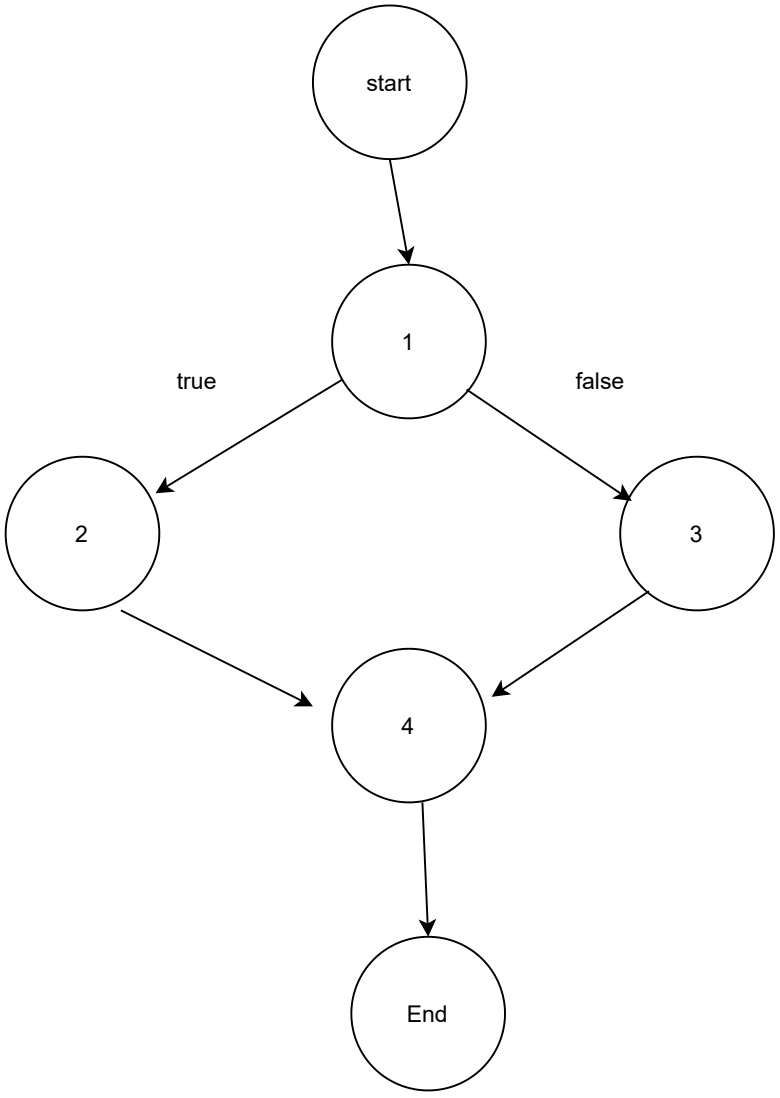
Block Number	Line Number	Entry	Exit
1	26,27	26	27
2	28	28	28
3	31,32	31	32
4	39	39	39



2. Open_token_stream

```
80    BufferedReader open_token_stream(String fname)
81    {
82        BufferedReader br;
83    if(fname==("null"))
84        br=open_character_stream(null);
85     else
86        br=open_character_stream(fname);
87     return br;
88    }
```

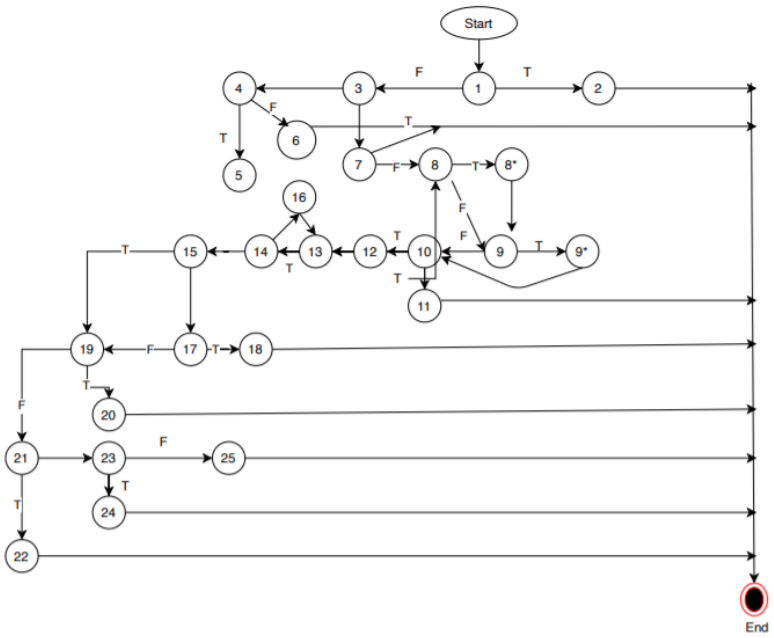
Block Number	Line Number	Entry	Exit
1	82,83	82	83
2	84	84	84
3	86	86	86
4	87	87	87



3. get_token

```
97   String get_token(BufferedReader br)
98   {
99     int i=0,j;
100    int id=0;
101    int res = 0;
102    char ch = '\0';
103
104    StringBuilder sb = new StringBuilder();
105
106     try {
107         res = get_char(br);
108         if (res == -1) {
109             return null;
110         }
111         ch = (char)res;
112        while(ch=='\t'||ch=='\n' || ch == '\r')     /* strip all blanks until meet characters */
113        {
114          res = get_char(br);
115          ch = (char)res;
116        }
117
118     if(res == -1)return null;
119     sb.append(ch);
120     if(is_spec_symbol(ch)==true)return sb.toString();
121     if(ch =='"')id=1;     /* prepare for string */
122     if(ch ==59)id=2;    /* prepare for comment */
123
124     res = get_char(br);
125     if (res == -1) {
126         unget_char(ch,br);
127         return sb.toString();
128     }
129     ch = (char)res;
130
```

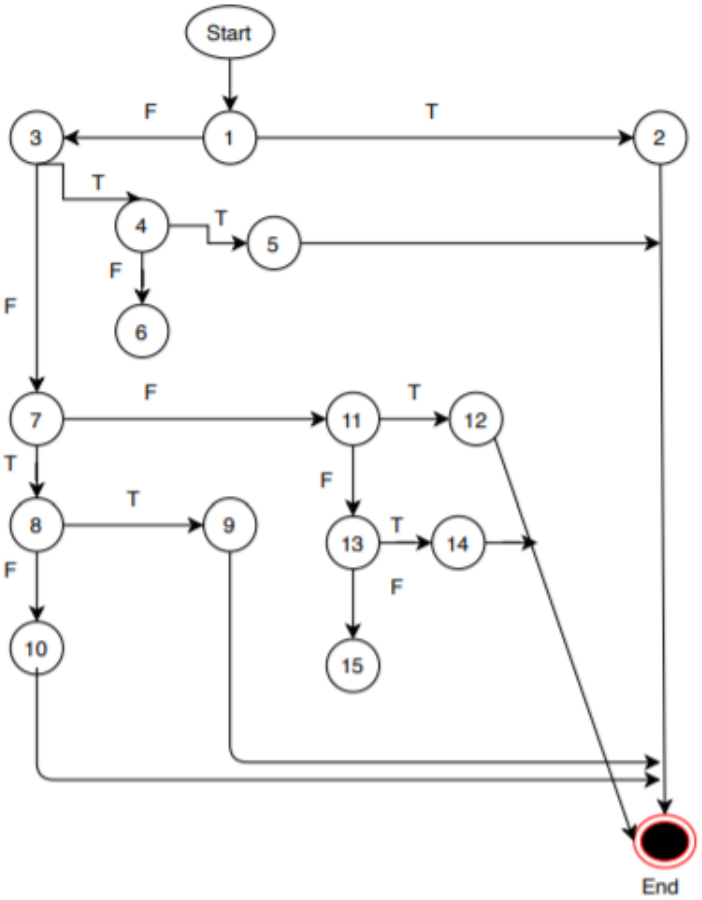
Block Number	Line Number	Entry	Exit
1	99,100,101,10	99	108
2	109	109	109
3	111	111	111
4	112	112	112
5	114,115	114	115
6	118	118	118
7	119,120	119	120
8	121	121	121
9	122	122	122
10	123,124,125	123	124
11	126,127	126	127
12	129	129	129
13	131	131	131
14	133,134,135,1	133	136
15	137	137	137
16	139	139	139
17	142	142	142
18	143,144	143	144
19	147	147	147
20	148,149	148	149
22	151	151	151
23	158	158	158
24	160,161	160	161
25	167	167	167



4. is_token_end

```
175    static boolean is_token_end(int str_com_id, int res)
176    {
177     if(res==-1)return(true); /* is eof token? */
178     char ch = (char)res;
179     if(str_com_id==1)          /* is string token */
180        { if(ch=='"' || ch=='\n' || ch == '\r')   /* for string until meet another " */
181             return true;
182          else
183             return false;
184        }
185
186     if(str_com_id==2)    /* is comment token */
187       { if(ch=='\n' || ch == '\r' || ch=='\t')     /* for comment until meet end of line */
188            return true;
189          else
190            return false;
191       }
192
193     if(is_spec_symbol(ch)==true) return true; /* is special_symbol? */
194     if(ch ==' ' || ch=='\n'|| ch=='\r' || ch==59) return true;
195                                  /* others until meet blank or tab or 59 */
196     return false;               /* other case,return FALSE */
197    }
```

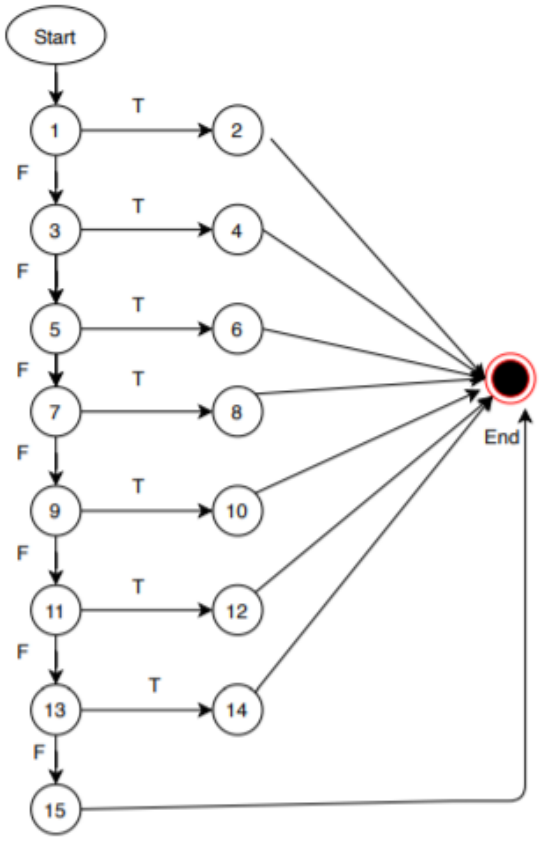
Block Number	Line Number	Entry	Exit
1	177	177	177
2	177	177	177
3	178,179	178	179
4	180	180	180
5	181	181	181
6	183	183	183
7	186	186	186
8	187	187	187
9	188	188	188
10	190	190	190
11	193	193	193
12	193	193	193
13	194	194	194
14	194	194	194
15	196	196	196



5. token_type

```
206    static int token_type(String tok)
207    {
208     if(is_keyword(tok))return(keyword);
209     if(is_spec_symbol(tok.charAt(0)))return(spec_symbol);
210     if(is_identifier(tok))return(identifier);
211     if(is_num_constant(tok))return(num_constant);
212     if(is_str_constant(tok))return(str_constant);
213     if(is_char_constant(tok))return(char_constant);
214     if(is_comment(tok))return(comment);
215     return(error);                    /* else look as error token */
216    }
```

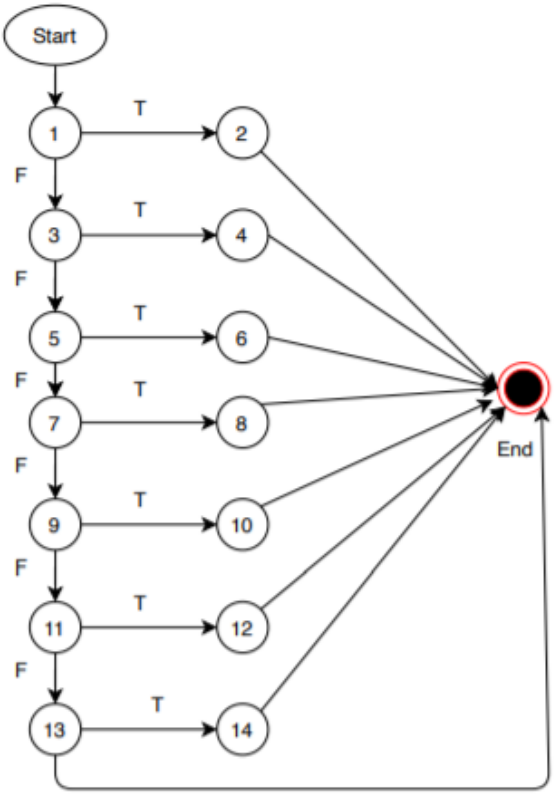
Block Number	Block Number	Block Number	Block Number
1	208	208	208
2	208	208	208
3	209	209	209
4	209	209	209
5	210	210	210
6	210	210	210
7	211	211	211
8	211	211	211
9	212	212	212
10	212	212	212
11	213	213	213
12	213	213	213
13	214	214	214
14	214	214	214
15	215	215	215



6. print_token

```
222    void print_token(String tok)
223    { int type;
224      type=token_type(tok);
225     if(type==error)
226       {
227        System.out.print("error,\"" + tok + "\".\n");
228       }
229
230     if(type==keyword)
231       {
232       System.out.print("keyword,\"" + tok + "\".\n");
233       }
234
235     if(type==spec_symbol)print_spec_symbol(tok);
236     if(type==identifier)
237       {
238       System.out.print("identifier,\"" + tok + "\".\n");
239       }
240     if(type==num_constant)
241       {
242       System.out.print("numeric," + tok + ".\n");
243       }
244     if(type==char_constant)
245       {
246        System.out.print("character,\"" + tok.charAt(1) + "\".\n");
247       }
248     if(type==comment)
249       {
250        System.out.print("comment,\"" + tok + "\".\n");
251       }
252   }
```

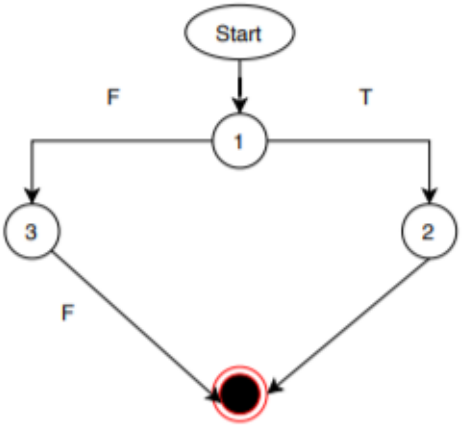
Block Number	Line Number	Entry	Exit
1	223,224,225	223	225
2	227	227	227
3	230	230	230
4	232	232	232
5	235	235	235
6	235	235	235
7	236	236	236
8	238	238	238
9	240	240	240
10	242	242	242
11	244	244	244
12	246	246	246
13	248	248	248
14	250	250	250



7. is_comment

```
262     static boolean is_comment(String ident)
263     {
264       if( ident.charAt(0) ==59 )   /* the char is 59   */
265          return true;
266       else
267          return false;
268     }
```

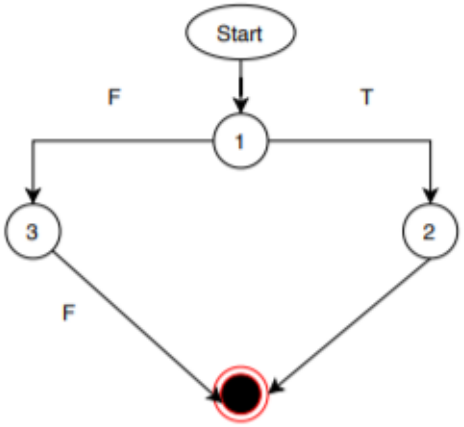
Block Number	Line Number	Entry	Exit
1	264	264	264
2	265	265	265
3	267	267	267



8. is_keyword

```
275    static boolean is_keyword(String str)
276    {
277     if (str.equals("and") || str.equals("or") || str.equals("if") ||
278             str.equals("xor")||str.equals("lambda")||str.equals("=>"))
279          return true;
280       else
281          return false;
282    }
```

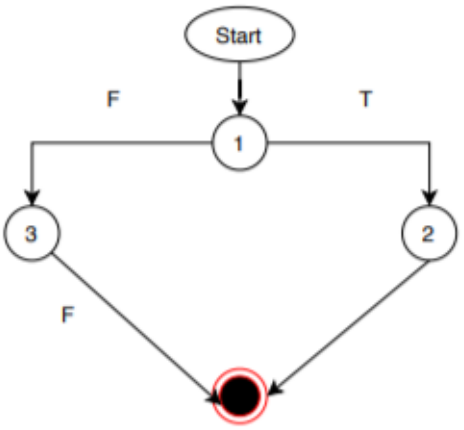
Block Number	Line Number	Entry	Exit
1	277,278	277	278
2	279	279	279
3	281	281	281



9. is_char_constant

```
289    static boolean is_char_constant(String str)
290    {
291       if (str.length() == 2 && str.charAt(0)=='#' && Character.isLetter(str.charAt(1)))
292         return true;
293      else
294         return false;
295    }
```

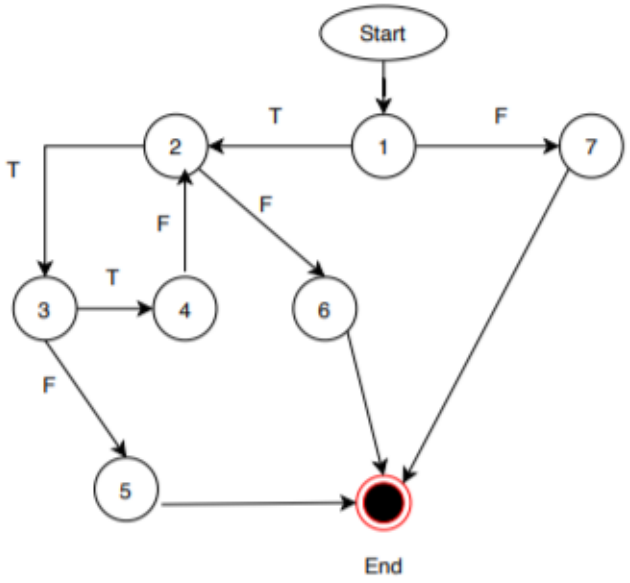
Block Number	Line Number	Entry	Exit
1	291	291	291
2	292	292	292
3	294	294	294



10. is_num_constant

```
302     static boolean is_num_constant(String str)
303     {
304       int i=1;
305
306       if ( Character.isDigit(str.charAt(0)))
307         {
308         while ( i < str.length() && str.charAt(i) != '\0' )   /* until meet token end sign */
309           {
310            if(Character.isDigit(str.charAt(i)))
311              i++;
312            else
313              return false;
314           }                          /* end WHILE */
315         return true;
316         }
317       else
318        return false;               /* other return FALSE */
319     }
```

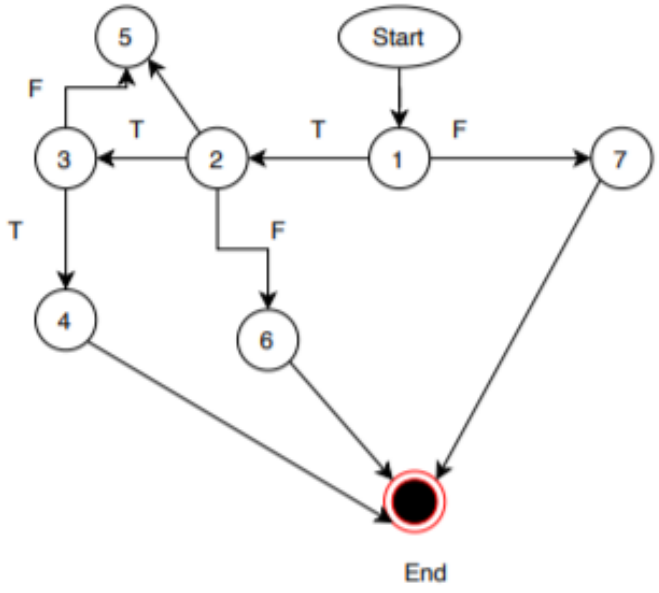
Block Number	Line Number	Entry	Exit
1	304,305	304	305
2	308	308	308
3	310	310	310
4	311	311	311
5	313	313	313
6	315	315	315
7	318	318	318



11. is_str_constant

```
326    static boolean is_str_constant(String str)
327    {
328      int i=1;
329
330      if ( str.charAt(0) =='"')
331         { while (i < str.length() && str.charAt(i)!='\0')  /* until meet the token end sign */
332             { if(str.charAt(i)=='"')
333                 return true;        /* meet the second '"'           */
334               else
335               i++;
336             }               /* end WHILE */
337        return false;
338        }
339      else
340        return false;       /* other return FALSE */
341    }
```

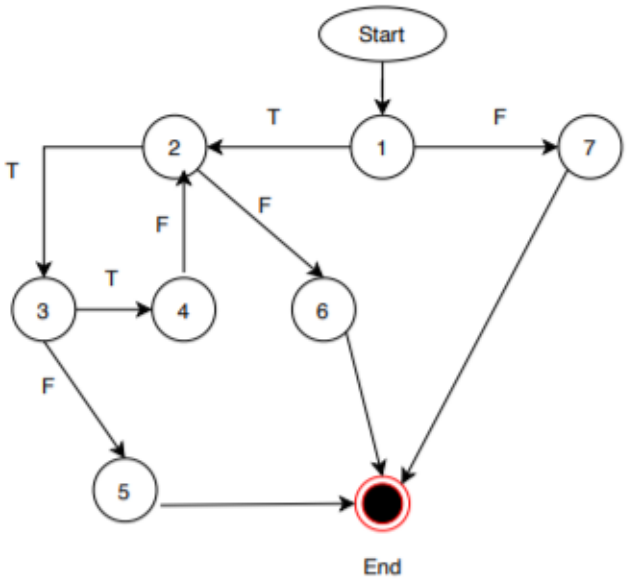
Block Number	Line Number	Entry	Exit
1	328, 330	328	330
2	331	331	331
3	332	332	332
4	333	333	333
5	335	335	335
6	337	337	337
7	340	340	340



12. is_identifier

```
348    static boolean is_identifier(String str)
349    {
350      int i=0;
351
352      if ( Character.isLetter(str.charAt(0)) )
353         {
354            while(i < str.length() && str.charAt(i) !='\0' )   /* unti meet the end token sign */
355               {
356                if(Character.isLetter(str.charAt(i)) || Character.isDigit(str.charAt(i)))
357                   i++;
358                else
359                   return false;
360               }      /* end WHILE */
361            return false;
362         }
363      else
364        return false;
```

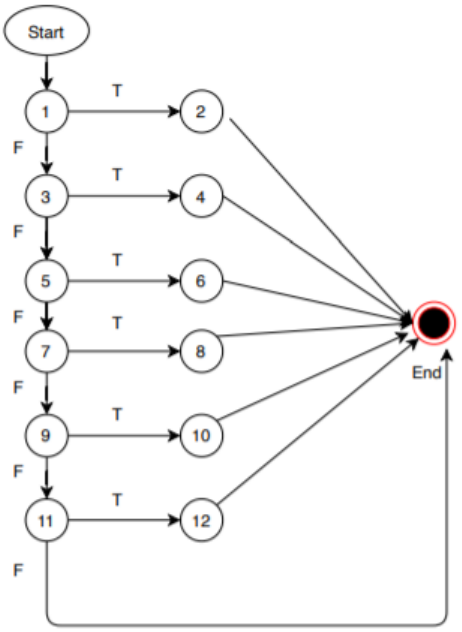
Block Number	Line Number	Entry	Exit
1	350,352	350	352
2	354	354	354
3	356	356	356
4	357	357	357
5	359	359	359
6	361	361	361
7	364	364	364



13. print_spec_symbol

```
375  static void print_spec_symbol(String str)
376  {
377      if      (str.equals("("))
378      {
379
380               System.out.print("lparen.\n");
381               return;
382      }
383      if (str.equals(")"))
384      {
385
386               System.out.print("rparen.\n");
387               return;
388      }
389      if (str.equals("["))
390      {
391               System.out.print("lsquare.\n");
392               return;
393      }
394      if (str.equals("]"))
395      {
396
397               System.out.print("rsquare.\n");
398               return;
399      }
400      if (str.equals("'"))
401      {
402               System.out.print("quote.\n");
403               return;
404      }
405      if (str.equals("`"))
406      {
407
408               System.out.print("bquote.\n");
409               return;
410      }
411
```

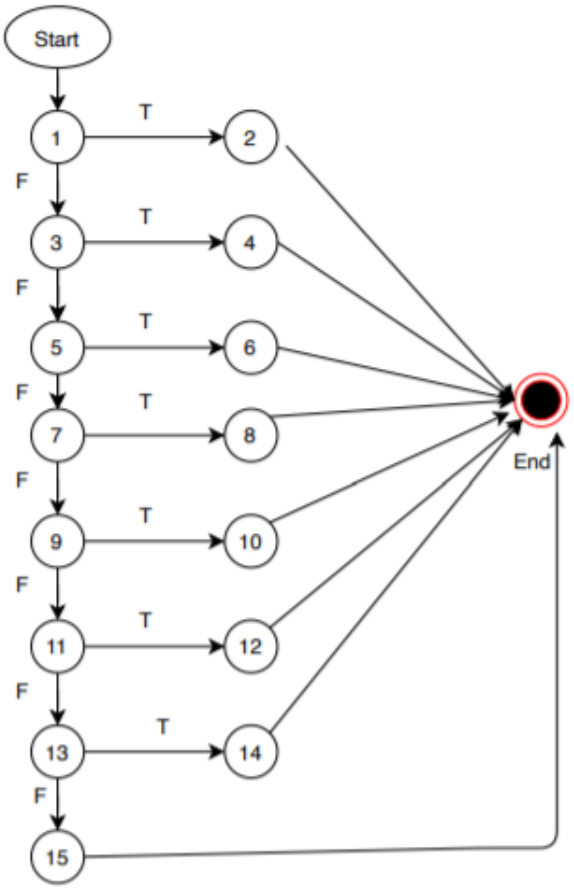
Block Number	Line Number	Entry	Exit
1	377	377	377
2	380,381	380	381
3	383	383	383
4	386,387	386	387
5	389	389	389
6	391,392	391	392
7	394	394	394
8	396 397	396	397
9	400	400	400
10	402	402	402
11	405	405	405
12	408 409	408	409



14. is_spec_symbol

```
420    static boolean is_spec_symbol(char c)
421    {
422        if (c == '(')
423        {
424            return true;
425        }
426        if (c == ')')
427        {
428            return true;
429        }
430        if (c == '[')
431        {
432            return true;
433        }
434        if (c == ']')
435        {
436            return true;
437        }
438        if (c == '"')
439        {
440            return true;
441        }
442        if (c == '`')
443        {
444            return true;
445        }
446        if (c == ',')
447        {
448            return true;
449        }
450        return false;     /* others return FALSE */
451    }
```

Block Number	Line Number	Entry	Exit
1	422	422	422
2	424	424	424
3	426	426	426
4	428	428	428
5	430	430	430
6	432	432	432
7	434	434	434
8	436	436	436
9	438	438	438
10	440	440	440
11	442	442	442
12	444	444	444
13	446	446	446
14	448	448	448
15	450	450	450



15. main

```
453     public static void main(String[] args)  {
454         String fname = null;
455         if (args.length == 0) { /* if not given filename,take as '"' */
456             fname = new String();
457 
458         } else if (args.length == 1) {
459             fname = args[0];
460         } else {
461             System.out.print("Error! Please give the token stream\n");
462             System.exit(0);
463         }
464         Printtokens t = new Printtokens();
465         BufferedReader br = t.open_token_stream(fname); /* open token stream */
466         String tok = t.get_token(br);
467         while (tok != null) {   /* take one token each time until eof */
468             t.print_token(tok);
469             //boolean actual_output = Printtokens.is_keyword(keyword);
470             //assertEquals(true, actual_output);
471             tok = t.get_token(br);
472 
473         }
474 
475         System.exit(0);
476     }
477 }
```

Block Number	Line Number	Entry	Exit
1	454,455	454	455
2	456	456	456
3	458	458	458
4	459	459	459
5	461,462	461	462
6	464,465,466	464	466
7	467	467	467
8	468,471	468	471
9	475	475	475