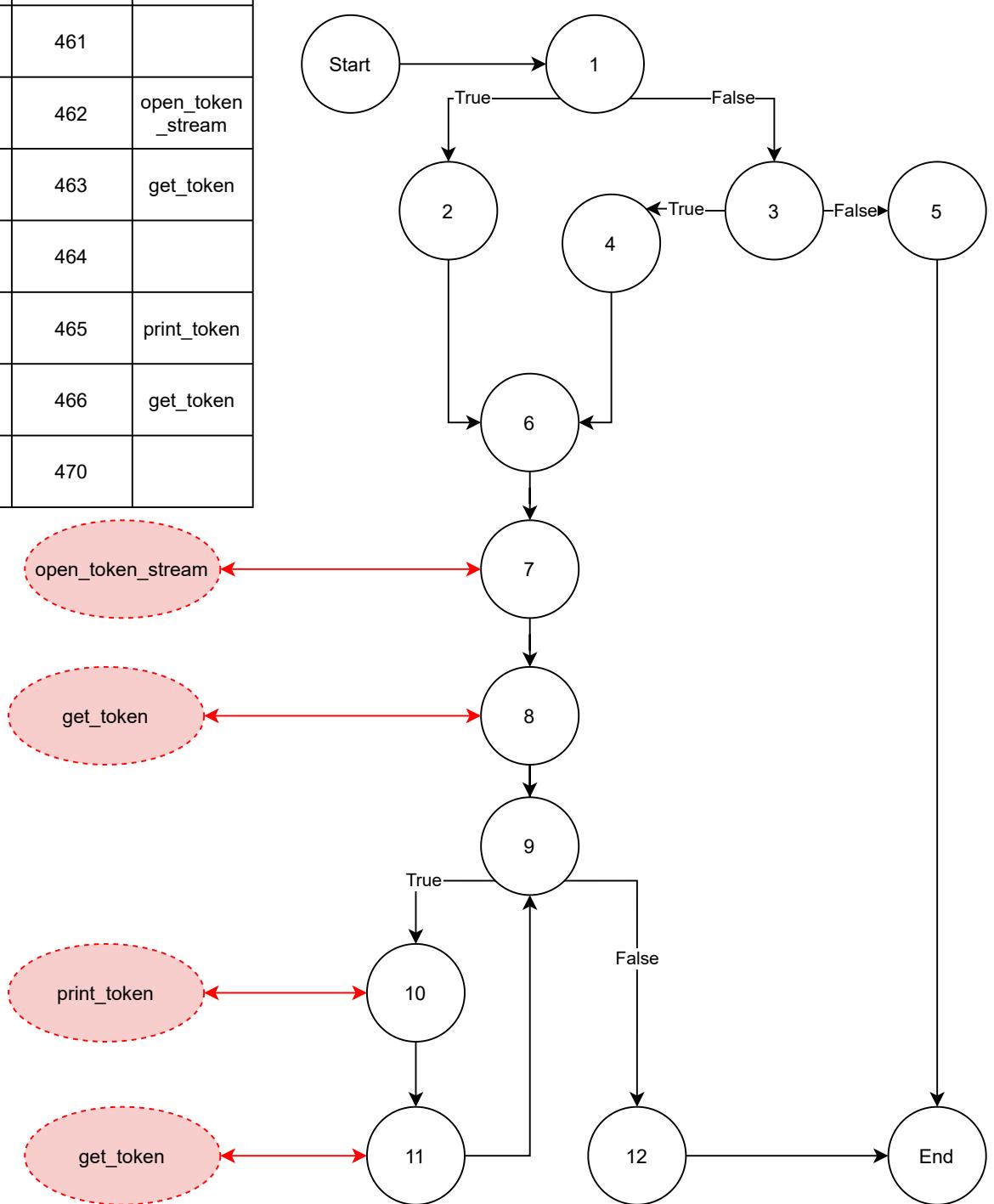


Main

Block No.	Lines	Entry	Exit	Func call
1	451, 452	451	452	
2	453	453	453	
3	455	455	455	
4	456	456	456	
5	458, 459	458	459	
6	461	461	461	
7	462	462	462	open_token_stream
8	463	463	463	get_token
9	464	464	464	
19	465	465	465	print_token
11	466	466	466	get_token
12	470	470	470	

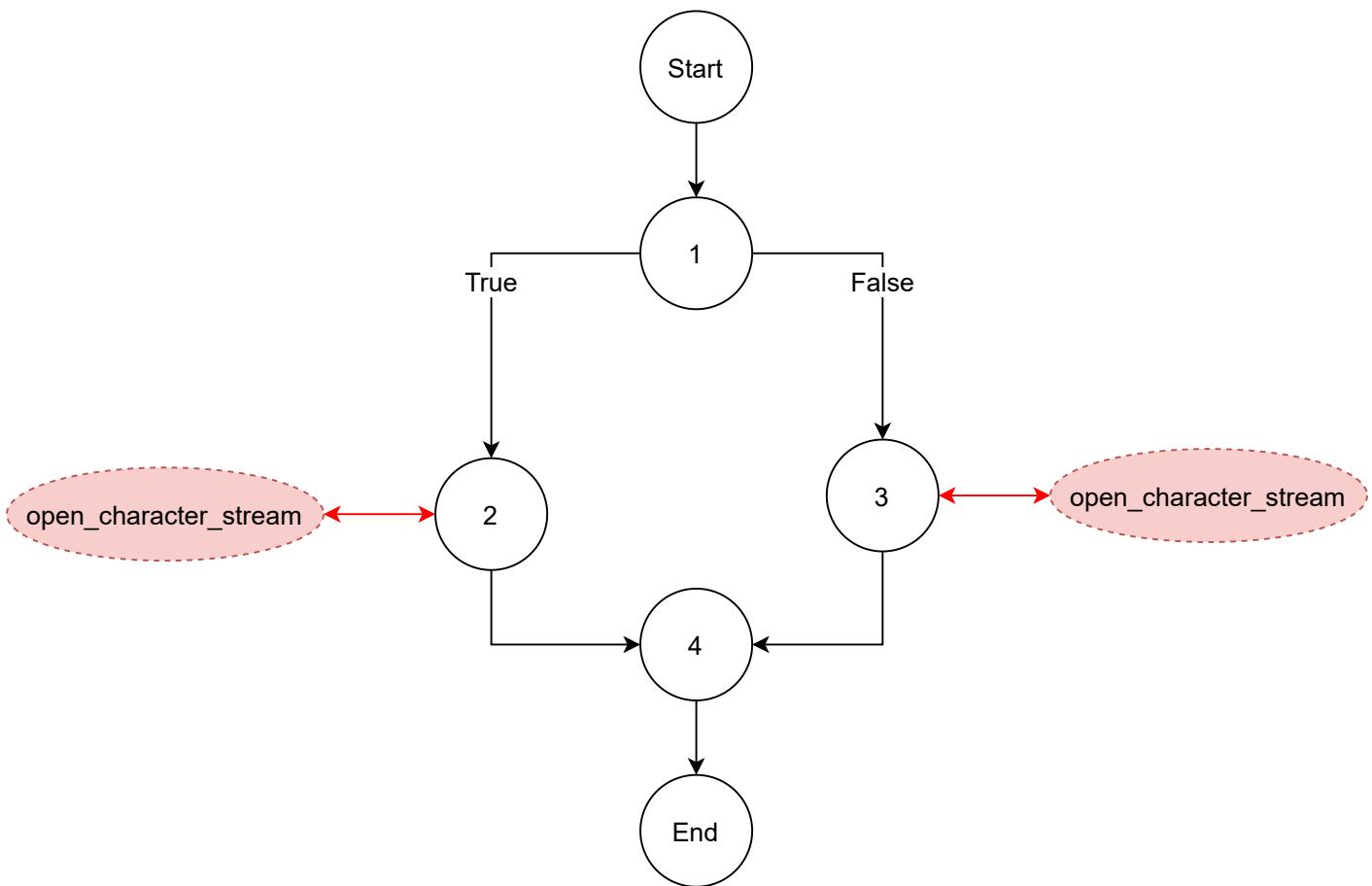
```
450● public static void main(String[] args) {
451     String fname = null;
452     if (args.length == 0) { /* if not given filename,take as ' ' */
453         fname = new String();
454     }
455     } else if (args.length == 1) {
456         fname = args[0];
457     } else {
458         System.out.print("Error!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



open_token_stream

```
77     BufferedReader open_token_stream(String fname)
78     {
79         BufferedReader br;
80         if(fname.equals(""))
81             br=open_character_stream(null);
82         else
83             br=open_character_stream(fname);
84         return br;
85     }
```

Block No	Lines	Entry	Exit	Func call
1	79, 80	79	80	
2	81	81	81	open_character_stream
3	83	83	83	open_character_stream
4	84	84	84	

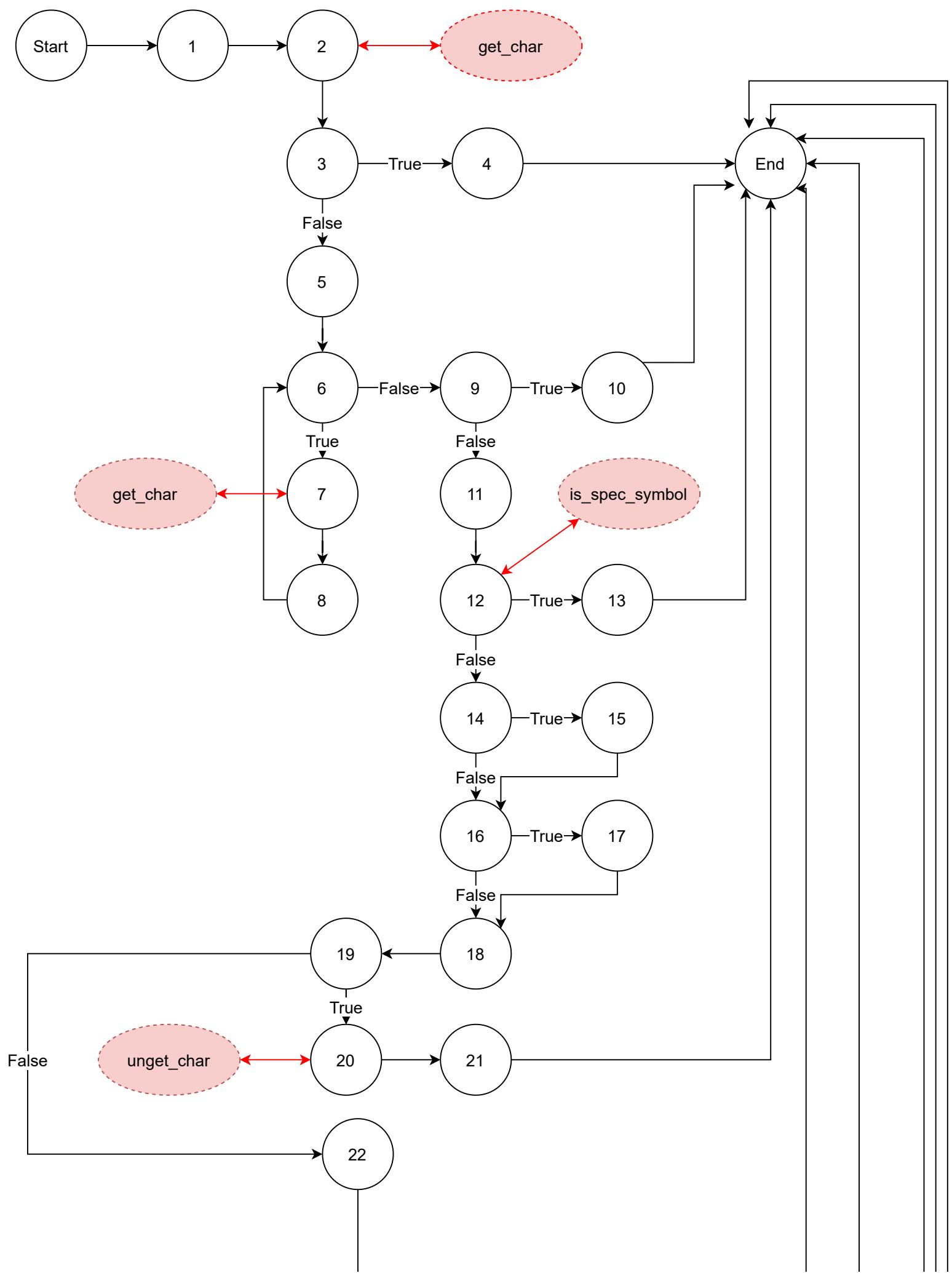


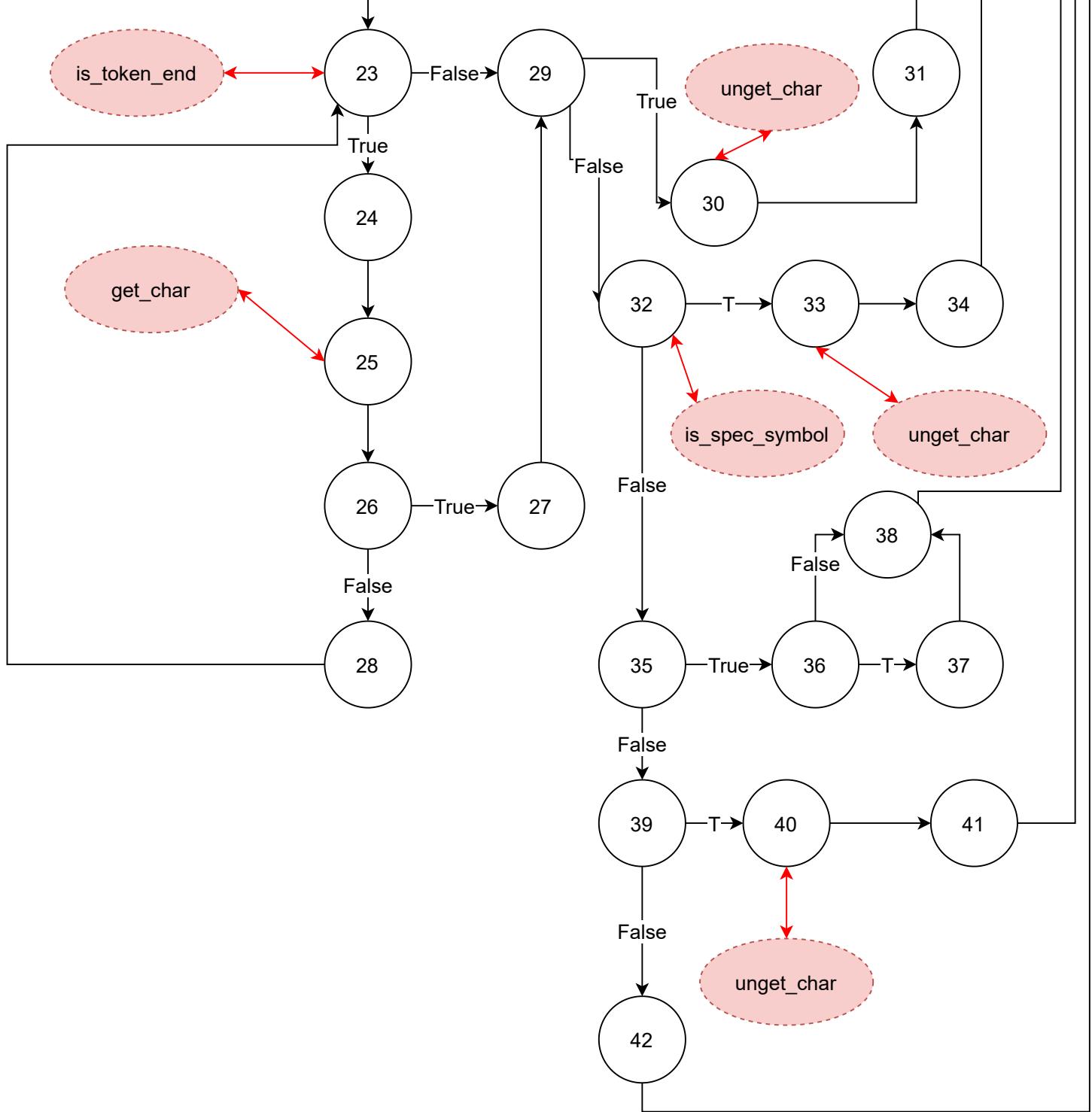
get_token

Block No	Lines	Entry	Exit	Func call
1	96,97,98,99, 101	96	101	
2	104	104	104	get_char
3	105	105	105	
4	106	106	106	
5	108	108	108	
6	109	109	109	
7	111	111	111	get_char
8	112	112	112	
9	115a	115a	115a	
10	115b	115b	115b	
11	116	116	116	
12	117a	117a	117b	is_spec_symbol
13	117b	117b	117b	
14	118a	118a	118a	
15	118b	118b	118b	
16	119a	119a	119a	
17	119b	119b	119b	
18	121	121	121	get_char
19	122	122	122	
20	123	123	123	unget_char
21	124	124	124	
22	126	126	126	

Block No	Lines	Entry	Exit	Func call
23	128	128	128	is_token_end
24	130, 131	130	131	
25	132	132	132	get_char
26	133	133	133	
27	134	134	134	
28	136	136	136	
29	139	139	139	
30	140	140	140	unget_char
31	141	141	141	
32	144	144	144	is_spec_symbol
33	145	145	145	unget_char
34	146	146	146	
35	148	148	148	
36	150	150	150	
37	151	151	151	
38	153	153	153	
39	155	155	155	
40	157	157	157	unget_char
41	158	158	158	
42	164	164	164	

```
94     String get_token(BufferedReader br)
95     {
96         int i=0,j;
97         int id=0;
98         int res = 0;
99         char ch = '\0';
100
101        StringBuilder sb = new StringBuilder();
102
103        try {
104            res = get_char(br);
105            if (res == -1) {
106                return null;
107            }
108            ch = (char)res;
109            while(ch=='\t'||ch=='\n' || ch == '\r')      /*
110            {
111                res = get_char(br);
112                ch = (char)res;
113            }
114
115            if(res == -1) return null;
116            sb.append(ch);
117            if(is_spec_symbol(ch)==true) return sb.toString();
118            if(ch == "'") id=2;      /* prepare for string */
119            if(ch == 59) id=1;      /* prepare for comment */
120
121            res = get_char(br);
122            if (res == -1) {
123                unget_char(ch,br);
124                return sb.toString();
125            }
126            ch = (char)res;
127
128            while (is_token_end(id,res) == false)/* until mee
129            {
130                sb.append(ch);
131                br.mark(4);
132                res = get_char(br);
133                if (res == -1) {
134                    break;
135                }
136                ch = (char)res;
137            }
138
139            if(res == -1)      /* if end of file */
140            { unget_char(ch,br);
141                return sb.toString();
142            }
143
144            if(is_spec_symbol(ch)==true)
145            { unget_char(ch,br);
146                return sb.toString();
147            }
148            if(id==1)          /* for string */
149            {
150                if (ch == "'") {
151                    sb.append(ch);
152                }
153                return sb.toString();
154            }
155            if(id==0 && ch==59)
156
157            { unget_char(ch,br);
158                return sb.toString();
159            }
160        } catch (IOException e) {
161            e.printStackTrace();
162        }
163
164        return sb.toString();
165    }
```

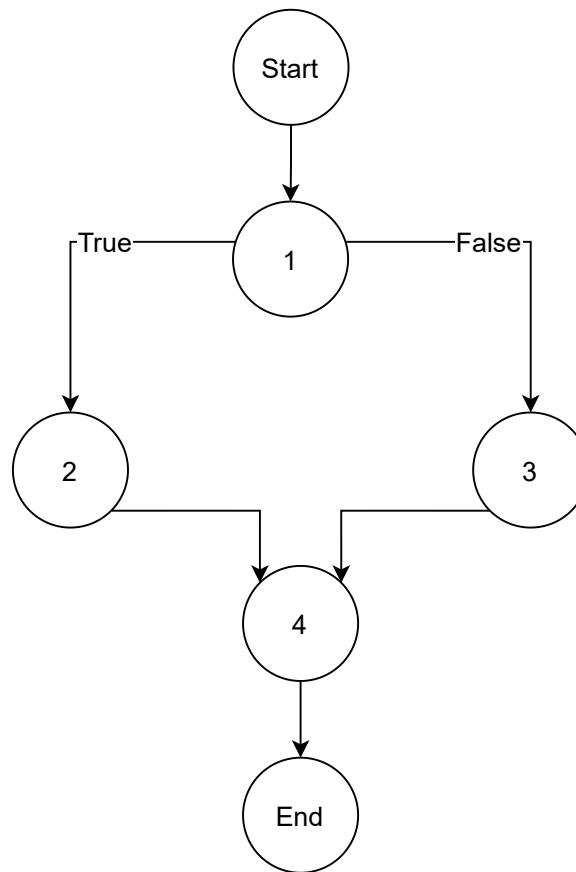




open_character_stream

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

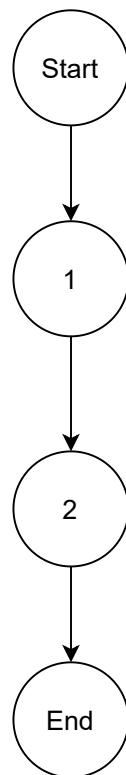
Block No	Lines	Entry	Exit	Func call
1	23, 24	23	24	
2	25	25	25	
3	28, 29	28	29	
4	36	36	36	



get_char

```
44●    int get_char(BufferedReader br){  
45        int ch = 0;  
46        try {  
47            br.mark(3);  
48            ch= br.read();  
49        } catch (IOException e) {  
50            e.printStackTrace();  
51        }  
52        return ch;  
53    }
```

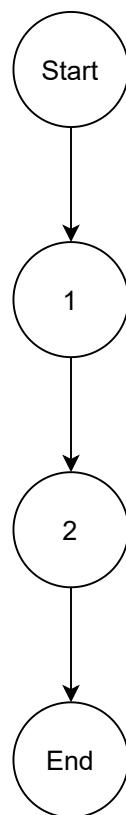
Block No	Lines	Entry	Exit	Func call
1	45, 47, 48	45	48	
2	52	52	52	



unget_char

```
61     char unget_char (int ch,BufferedReader br) {  
62         try {  
63             br.reset();  
64         } catch (IOException e) {  
65             e.printStackTrace();  
66         }  
67         return 0;  
68     }  
69 }
```

Block no	Lines	Entry	Exit	func call
1	63	63	63	
2	67	67	67	

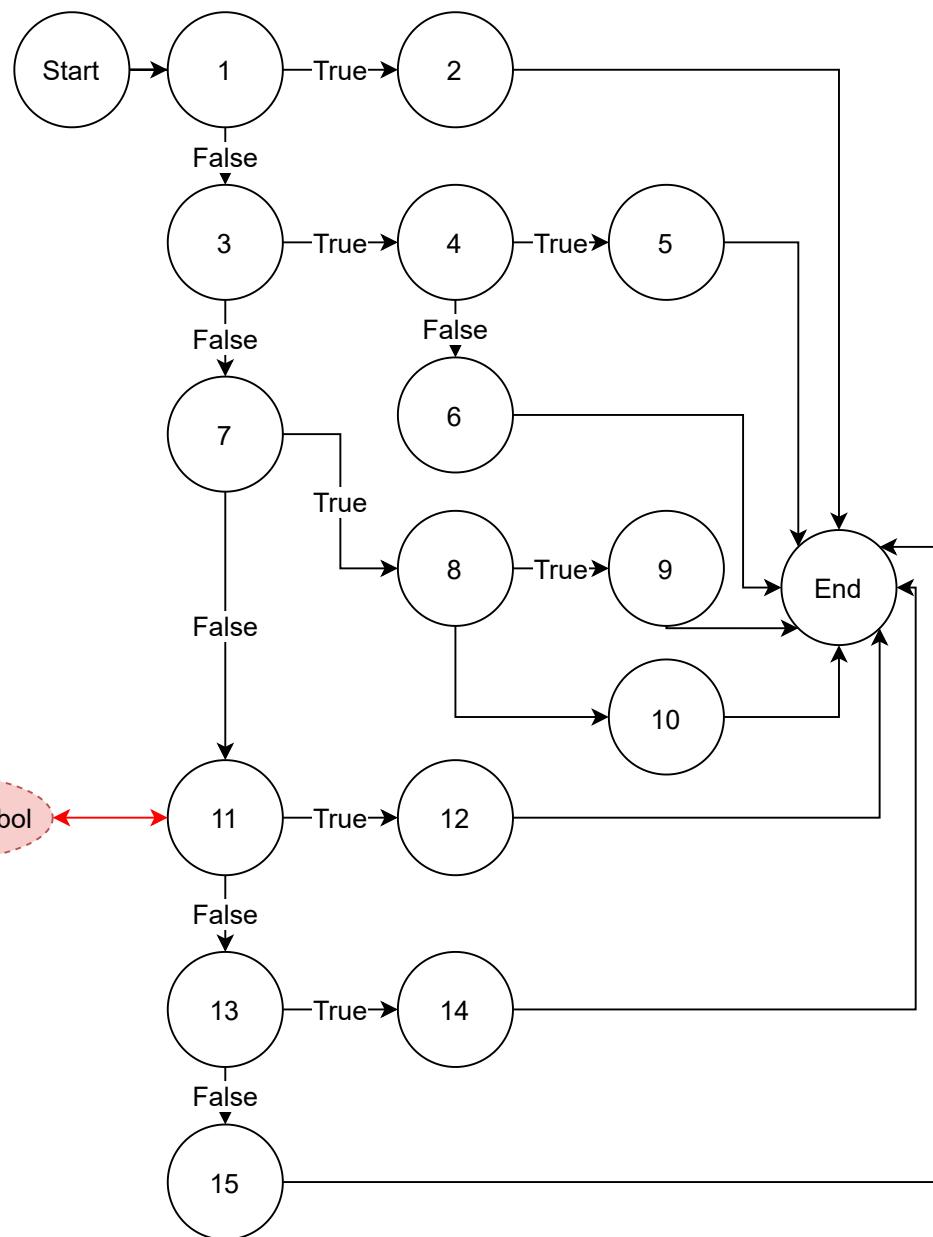


is_token_end

Block No	Lines	Entry	Exit	Func call
1	174a	174a	174a	
2	174b	174b	174b	
3	175, 176	175	176	
4	177	177	177	
5	178	178	178	
6	180	180	180	
7	183	183	183	
8	184	184	184	
9	185	185	185	
10	187	187	187	
11	190a	190a	190a	is_spec_symbol
12	190b	190b	190b	
13	191a	191a	191a	
14	191b	191b	191b	
15	193	193	193	

```

172 static boolean is_token_end(int str_com_id, int res)
173 {
174     if(res===-1) return(true); /* is eof token? */
175     char ch = (char)res;
176     if(str_com_id==1) /* is string token */
177     { if(ch==='"' || ch=='\n' || ch == '\r') /* for string
178         return true;
179     else
180         return false;
181     }
182
183     if(str_com_id==2) /* is comment token */
184     { if(ch=='\n' || ch == '\r' || ch=='\t') /* for comment
185         return true;
186     else
187         return false;
188     }
189
190     if(is_spec_symbol(ch)==true) return true; /* is special_symbol
191     if(ch == ' ' || ch=='\n'|| ch=='\r' || ch==59) return true;
192                                         /* others until meet blank or
193                                         /* other case,return FALSE */
194 }
```



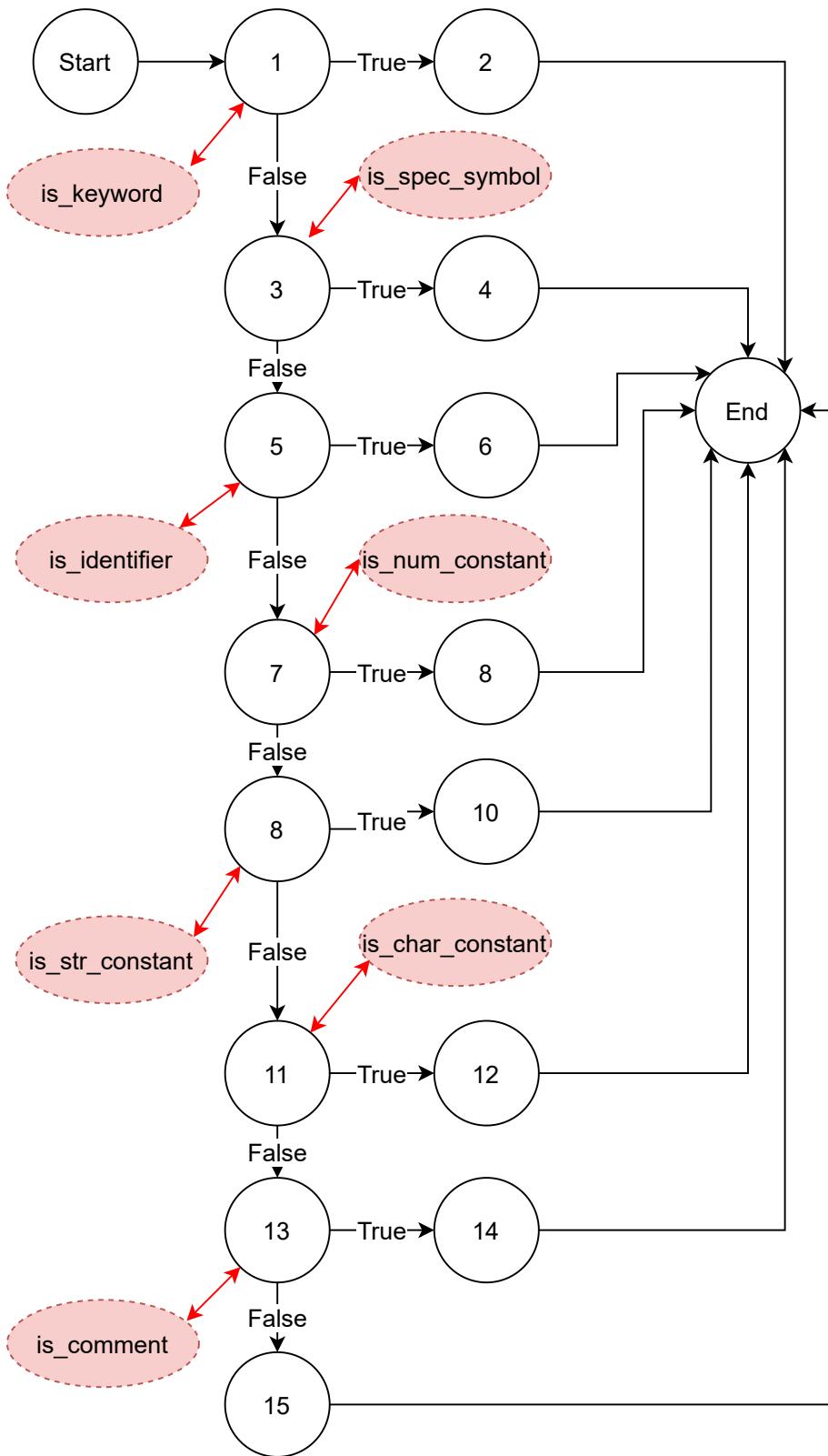
token_type

Block	Lines	Entry	Exit	func call
1	205a	205a	205a	is_keyword
2	205b	205b	205b	
3	206a	206a	206a	is_spec_symbol
4	206b	206b	206b	
5	207a	207a	207a	is_identifier
6	207b	207b	207b	
7	208a	208a	208a	is_num_constant
8	208b	208b	208b	
9	209a	209a	209a	is_str_constant
10	209b	209b	209b	
11	210a	210a	210a	is_char_consant
12	210b	210b	210b	
13	211a	211a	211a	is_comment
14	211b	211b	211b	
15	212	212	212	

```

203 static int token_type(String tok)
204 {
205     if(is_keyword(tok))return(keyword);
206     if(is_spec_symbol(tok.charAt(0)))return(spec_symbol);
207     if(is_identifier(tok))return(identifier);
208     if(is_num_constant(tok))return(num_constant);
209     if(is_str_constant(tok))return(str_constant);
210     if(is_char_constant(tok))return(char_constant);
211     if(is_comment(tok))return(comment);
212     return(error);
213 } /* else look as error

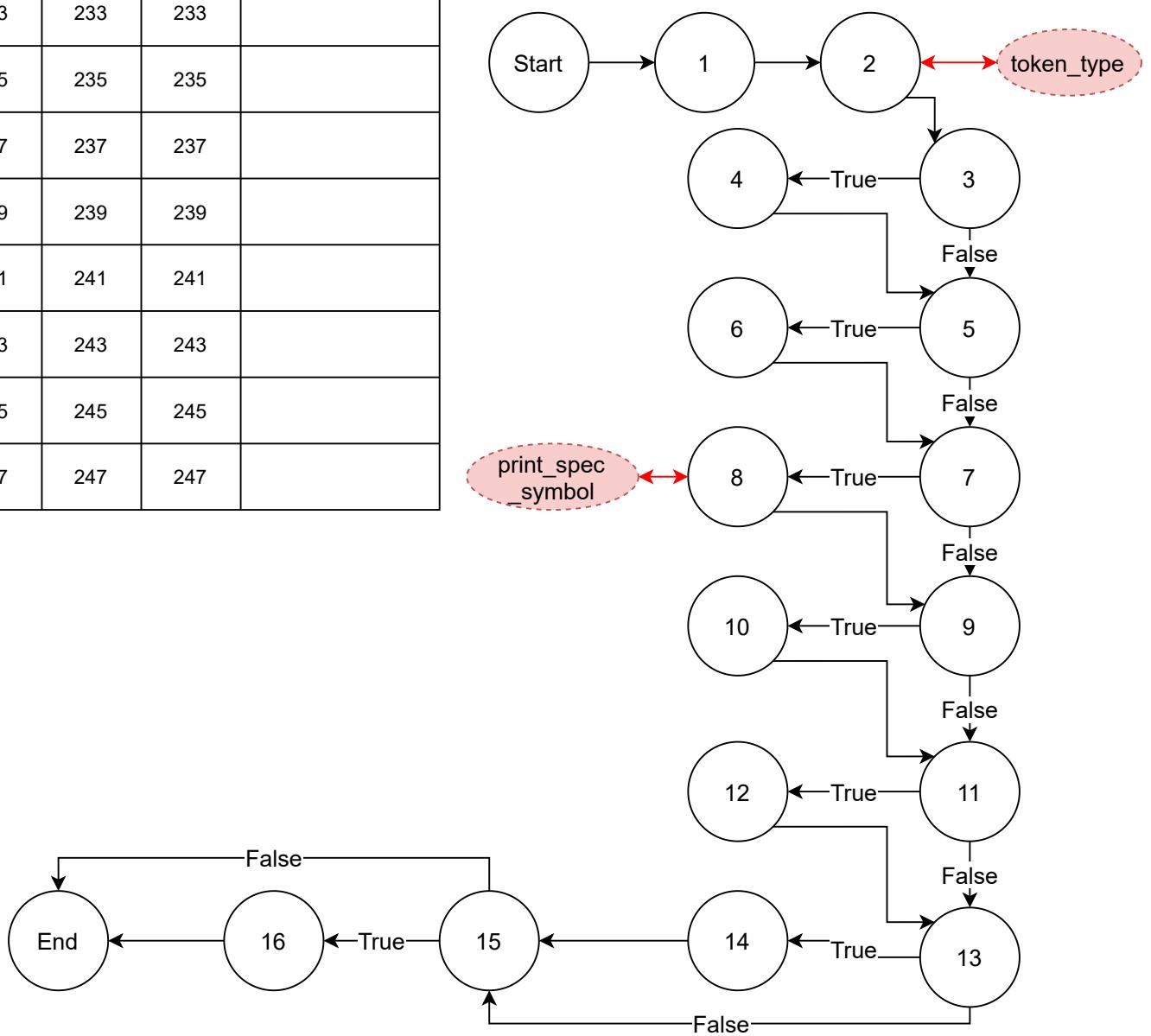
```



print_token

Block	Lines	Entry	Exit	Func call
1	220	220	220	
2	221	221	221	token_type
3	222	222	222	
4	224	224	224	
5	227	227	227	
6	229	229	229	
7	232a	232a	232a	
8	232b	232b	232b	print_spec_symbol
9	233	233	233	
10	235	235	235	
11	237	237	237	
12	239	239	239	
13	241	241	241	
14	243	243	243	
15	245	245	245	
16	247	247	247	

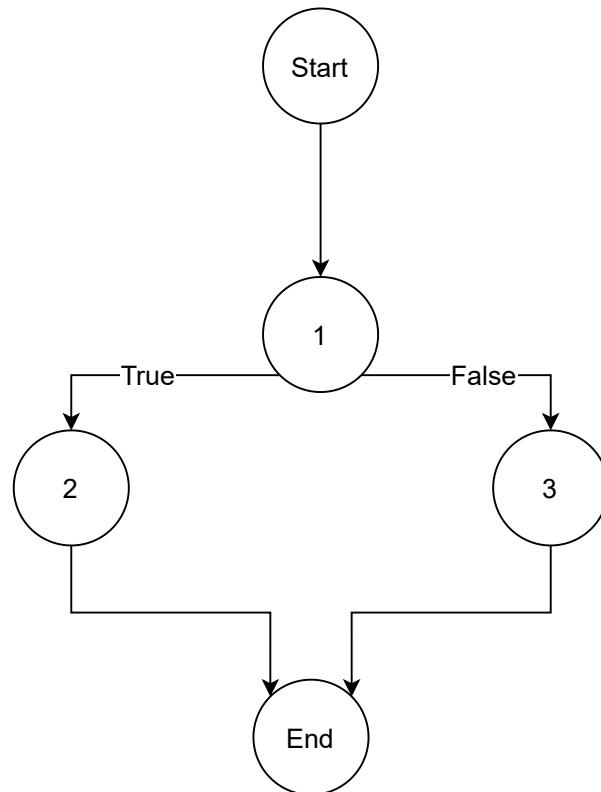
```
219● void print_token(String tok)
220 { int type;
221     type=token_type(tok);
222     if(type==error)
223     {
224         System.out.print("error,\\" + tok + "\.\n");
225     }
226
227     if(type==keyword)
228     {
229         System.out.print("keyword,\\" + tok + "\.\n");
230     }
231
232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



is_comment

```
259 static boolean is_comment(String ident)
260 {
261     if( ident.charAt(0) ==59 ) /* the ch
262         return true;
263     else
264         return false;
265 }
```

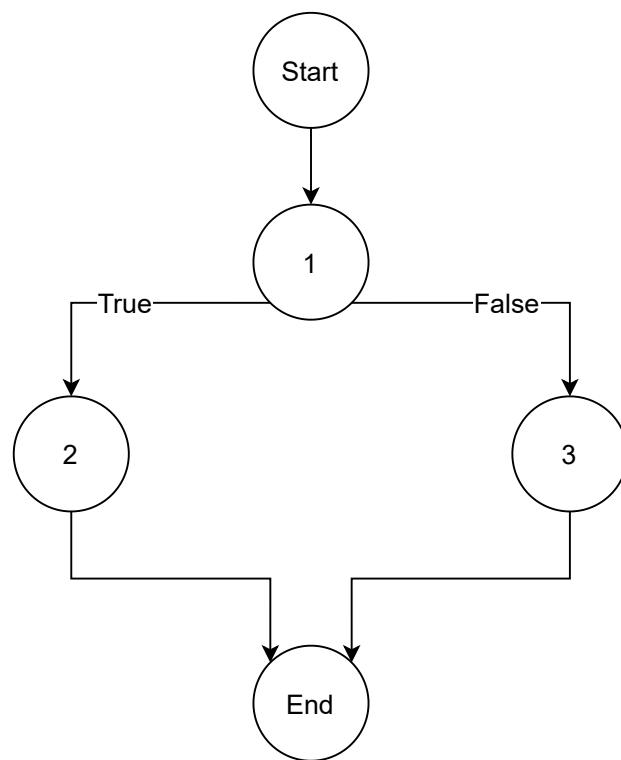
Block	Lines	Entry	Exit	Func call
1	261	261	261	
2	262	262	262	
3	264	264	264	



is_keyword

```
272     static boolean is_keyword(String str)
273     {
274         if (str.equals("and") || str.equals("or") || str.equals("if") ||
275             str.equals("xor")||str.equals("lambda")||str.equals("=>"))
276             return true;
277         else
278             return false;
279     }
```

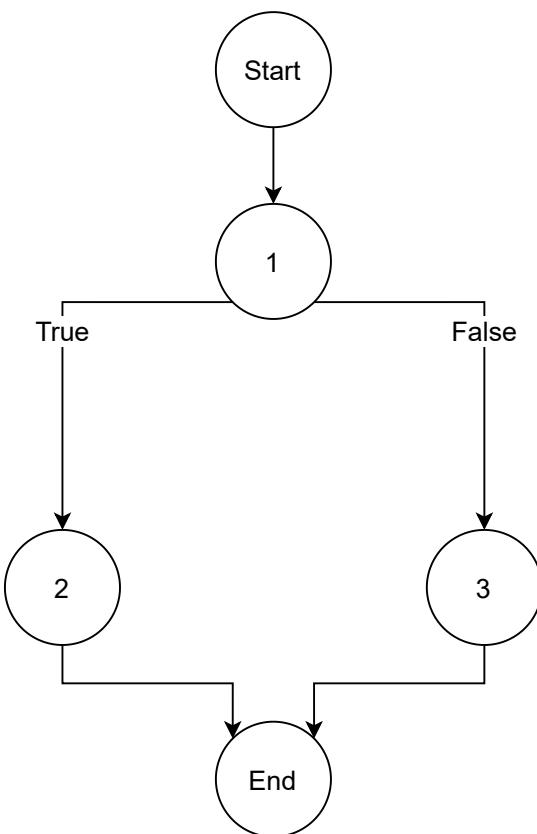
Block	Lines	Entry	Exit	func call
1	274, 275	274	275	
2	276	276	276	
3	278	278	278	



is_char_constant

```
286     static boolean is_char_constant(String str)
287     {
288         if (str.length() > 2 && str.charAt(0)=='#' && Character.isLetter(str.charAt(1)))
289             return true;
290         else
291             return false;
292     }
```

Block	Lines	Entry	Exit	Func call
1	288	288	288	
2	289	289	289	
3	291	291	291	

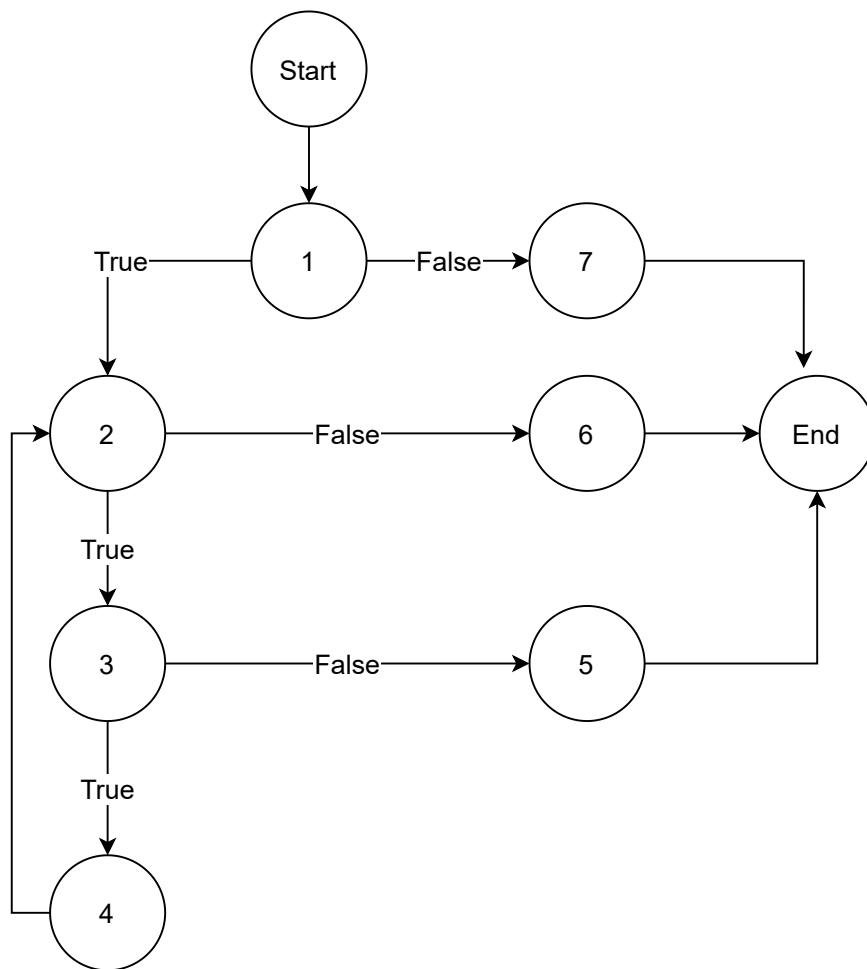


is_num_constant

Block	Lines	Entry	Exit	Func call
1	301, 303	301	303	
2	305	305	305	
3	307	307	307	
4	308	308	308	
5	310	310	310	
6	312	312	312	
7	315	315	315	

```

299 static boolean is_num_constant(String str)
300 {
301     int i=1;
302
303     if ( Character.isDigit(str.charAt(0)))
304     {
305         while ( i <= str.length() && str.charAt(i) != '\0' )
306         {
307             if(Character.isDigit(str.charAt(i+1)))
308                 i++;
309             else
310                 return false;
311         }                                /* end WHILE */
312     }
313     else
314         return false;                  /* other return FALSE */
315 }
316 }
```

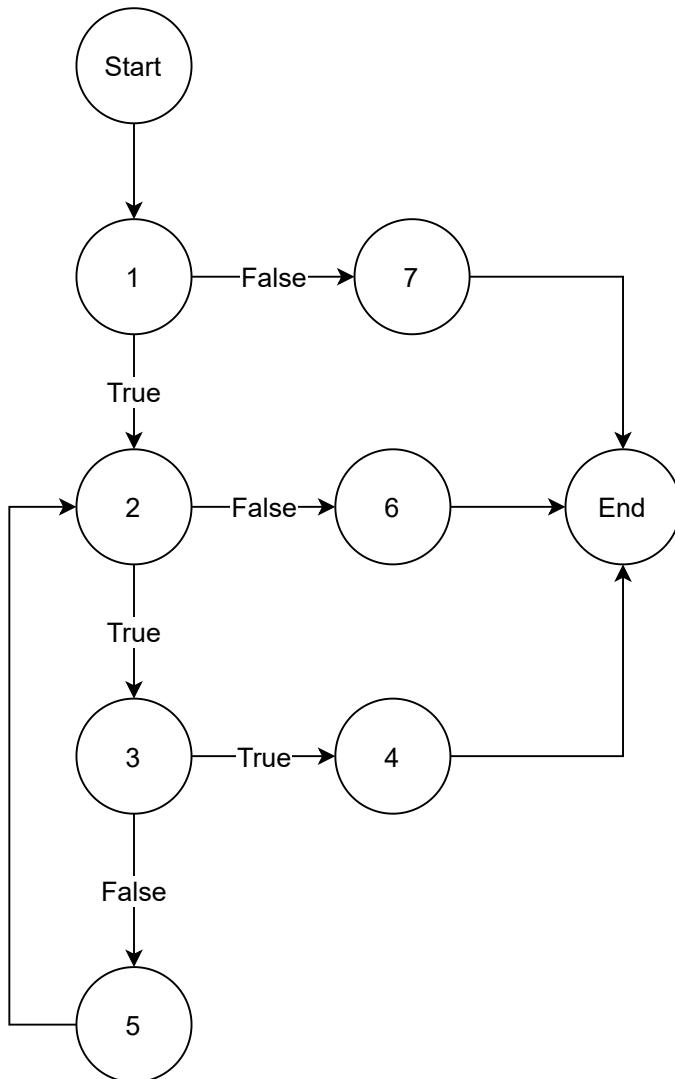


is_str_constant

Block	Lines	Entry	Exit	func call
1	325, 327	325	327	
2	328	328	328	
3	329	329	329	
4	330	330	330	
5	332	332	332	
6	334	334	334	
7	337	337	337	

```

323 static boolean is_str_constant(String str)
324 {
325     int i=1;
326
327     if ( str.charAt(0) =='"' )
328     { while ( i < str.length() && str.charAt(i)!='\0' )
329         { if(str.charAt(i)=='"')
330             return true; /* meet the second """
331         else
332             i++;
333         } /* end WHILE */
334     }
335     else
336         return false; /* other return FALSE */
338 }
```

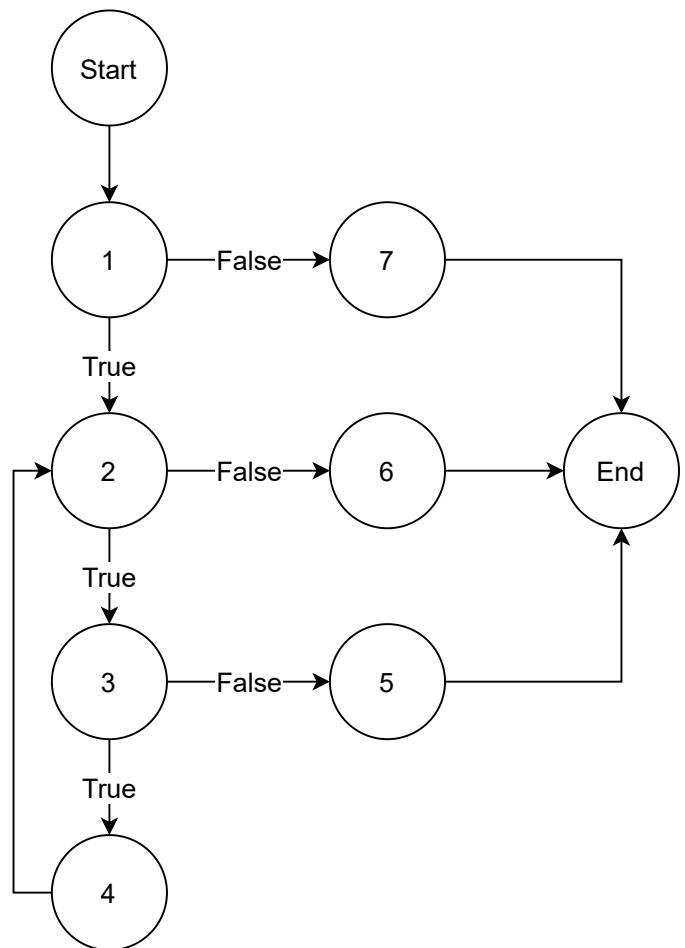


is_identifier

```

345● static boolean is_identifier(String str)
346 {
347     int i=0;
348
349     if ( Character.isLetter(str.charAt(0)) )
350     {
351         while(i < str.length() && str.charAt(i) !='\0' ) /* until meet the end token
352         {
353             if(Character.isLetter(str.charAt(i)) || Character.isDigit(str.charAt(i)))
354                 i++;
355             else
356                 return false;
357             } /* end WHILE */
358         return false;
359     }
360     else
361         return true;
362 }
```

Block	Lines	Entry	Exit	func call
1	347, 349	347	349	
2	351	351	351	
3	353	353	353	
4	354	354	354	
5	356	356	356	
6	358	358	358	
7	361	361	361	

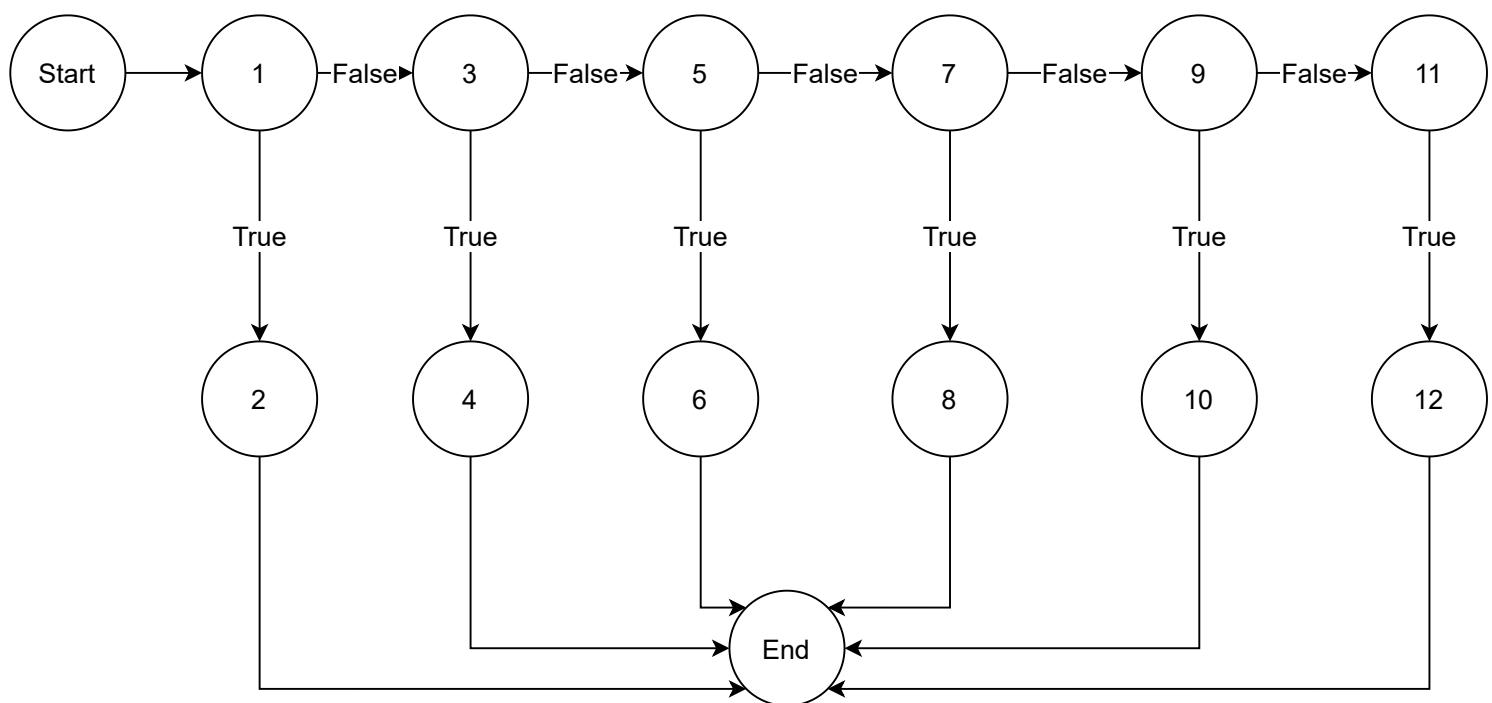


print_spec_symbol

Block	Lines	Entry	Exit	Func call
1	374	374	374	
2	377, 378	377	378	
3	380	380	380	
4	383, 384	383	384	
5	386	386	386	
6	388, 389	388	389	
7	391	391	391	
8	394, 395	394	395	
9	397	397	397	
10	399, 400	399	400	
11	402	402	402	
12	405, 406	405	406	

```

372     static void print_spec_symbol(String str)
373     {
374         if (str.equals("{"))
375         {
376             System.out.print("lparen.\n");
377             return;
378         }
379         if (str.equals("}"))
380         {
381             System.out.print("rparen.\n");
382             return;
383         }
384         if (str.equals("["))
385         {
386             System.out.print("lsquare.\n");
387             return;
388         }
389         if (str.equals("]"))
390         {
391             System.out.print("rsquare.\n");
392             return;
393         }
394         if (str.equals("\""))
395         {
396             System.out.print("quote.\n");
397             return;
398         }
399         if (str.equals("\`"))
400         {
401             System.out.print("bquote.\n");
402             return;
403         }
404     }
405 }
```



is_spec_symbol

Block	Lines	Entry	Exit	Func call
1	419	419	419	
2	421	421	421	
3	423	423	423	
4	425	425	425	
5	427	427	427	
6	429	429	429	
7	431	431	431	
8	433	433	433	
9	435	435	435	
10	437	437	437	
11	439	439	439	
12	441	441	441	
13	443	443	443	
14	445	445	445	
14	445	445	445	
15	447	447	447	

```

417 static boolean is_spec_symbol(char c)
418 {
419     if (c == '(')
420     {
421         return true;
422     }
423     if (c == ')')
424     {
425         return true;
426     }
427     if (c == '[')
428     {
429         return true;
430     }
431     if (c == ']')
432     {
433         return true;
434     }
435     if (c == '/')
436     {
437         return true;
438     }
439     if (c == '\\')
440     {
441         return true;
442     }
443     if (c == ',')
444     {
445         return true;
446     }
447     return false; /* others return false */
448 }
```

