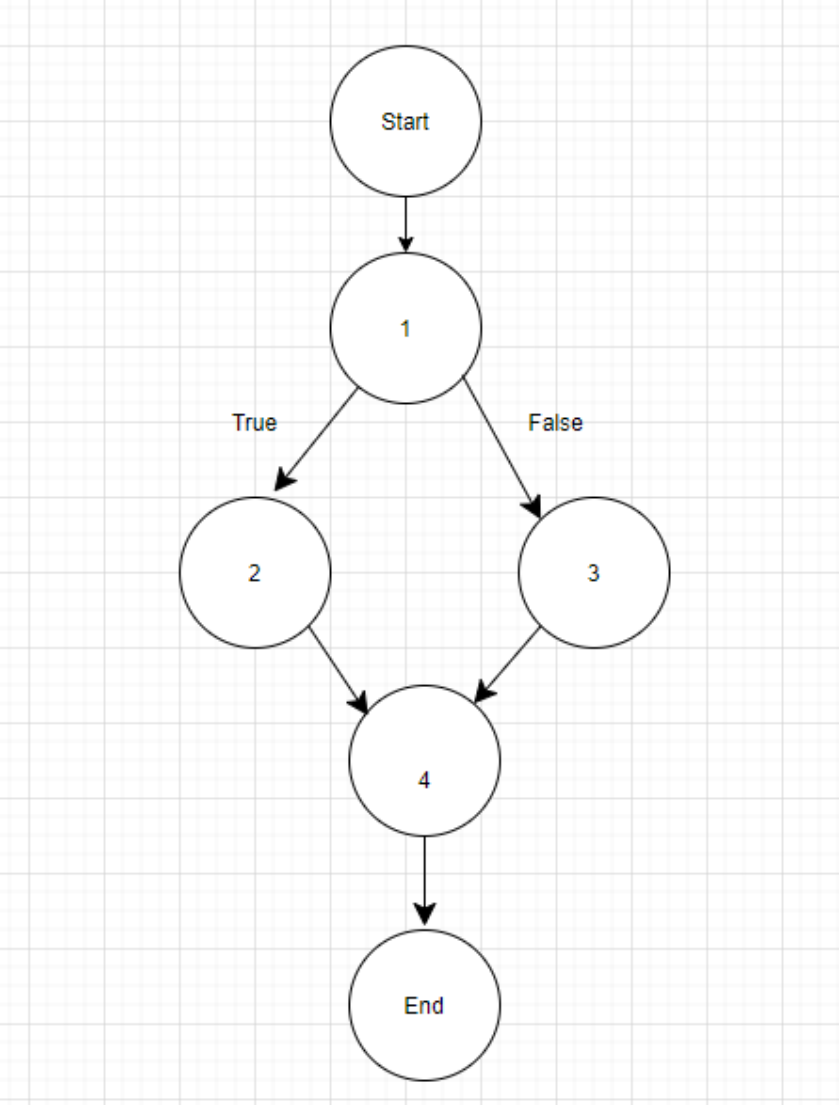
Sindhu parajuli
1001764819

Source code with block table and the CFG

1. open_character_stream

```
21    /                                                    /
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    }
```

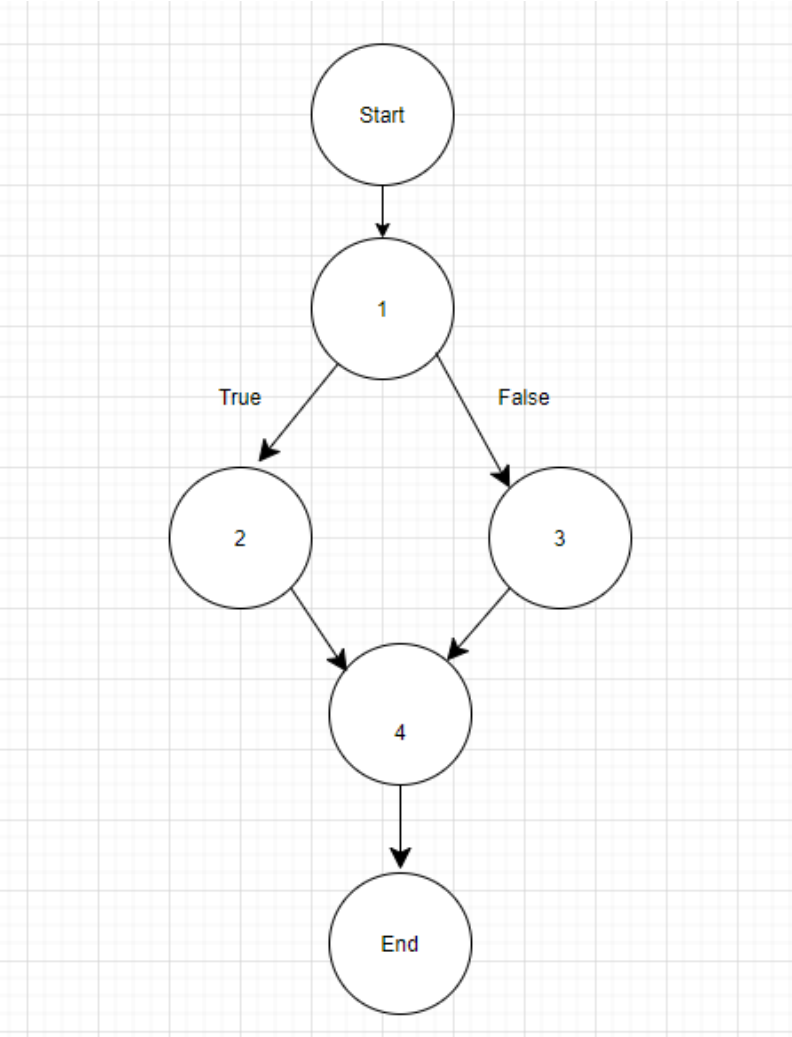
Block	Lines	Entry	Exit
1	23,24	23	24
2	25	25	25
3	28,29	28	29
4	36	36	36



2. 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      }
86
```

Block	Lines	Entry	Exit
1	79,80	79	80
2	81	81	81
3	83	83	83
4	84	84	84

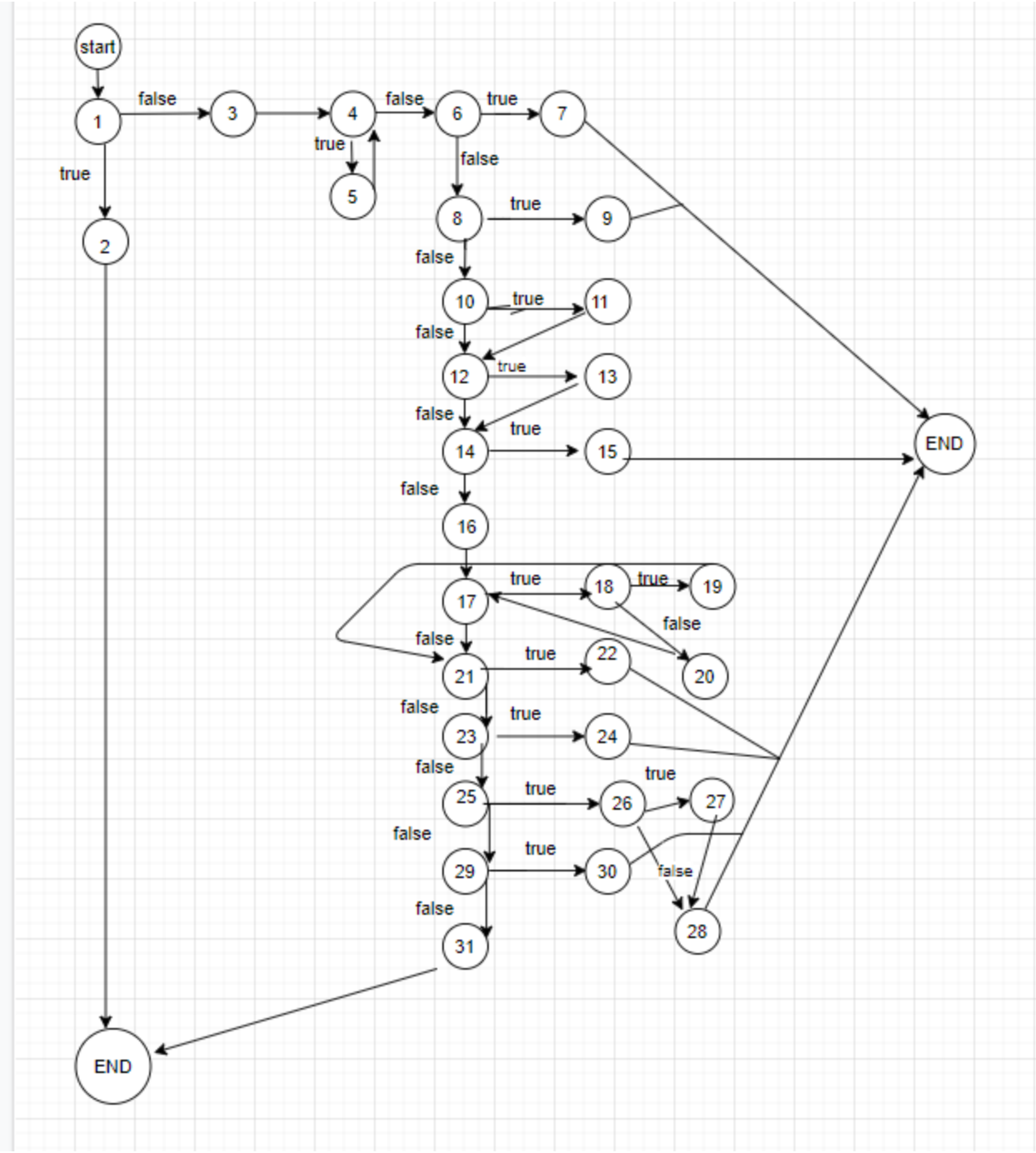


3. get_token

```
String get_token(BufferedReader br)
{
  int i=0,j;
  int id=0;
  int res = 0;
  char ch = '\0';

  StringBuilder sb = new StringBuilder();

   try {
       res = get_char(br);
       if (res == -1) {
           return null;
       }
       ch = (char)res;
      while(ch=='\t'||ch=='\n' || ch == '\r')     /* strip all blanks until meet characte
      {
        res = get_char(br);
        ch = (char)res;
      }

   if(res == -1)return null;
   sb.append(ch);
   if(is_spec_symbol(ch)==true)return sb.toString();
   if(ch =='"')id=2;    /* prepare for string */
   if(ch ==59)id=1;    /* prepare for comment */

   res = get_char(br);
   if (res == -1) {
       unget_char(ch,br);
       return sb.toString();
   }
   ch = (char)res;

   while (is_token_end(id,res) == false)/* until meet the end character */
   {
       sb.append(ch);
       br.mark(4);
       res = get_char(br);
       if (res == -1) {
           break;
       }
       ch = (char)res;
   }

   if(res == -1)       /* if end character is eof token    */
      { unget_char(ch,br);        /* then put back eof on token_stream */
        return sb.toString();
      }

   if(is_spec_symbol(ch)==true)     /* if end character is special_symbol */
      { unget_char(ch,br);        /* then put back this character       */
        return sb.toString();
      }
   if(id==1)                  /* if end character is " and is string */
     {
       if (ch == '"') {
           sb.append(ch);
       }
       return sb.toString();
     }
   if(id==0 && ch==59)
                                   /* when not in string or commen
     { unget_char(ch,br);        /* then put back this character
       return sb.toString();
     }
} catch (IOException e) {
    e.printStackTrace();
}

   return sb.toString();                    /* return nomal case to
}
```

Block	Lines	Entry	Exit
1	96,97,98,99,101,104,105	96	105
2	106	106	106
3	108	108	108
4	109	109	109
5	111,112	111	112
6	115a	115a	115a
7	115b	115b	115b
8	116,117a	116	117a
9	117b	117b	117b
10	118a	118a	118a
11	118b	118b	118b
12	119a	119a	119a
13	119b	119b	119b
14	121,122	121	122
15	123,124	123	124

16	126	126	126
17	128	128	128
18	130,131,132,133	130	133
19	134	134	134
20	136	136	136
21	139	139	139
22	140,141	140	141
23	144	144	144
24	145,146	145	146
25	148	148	148
26	150	150	150
27	151	151	151
28	153	153	153
29	155	155	155
30	157,158	157	158
31	164	164	164

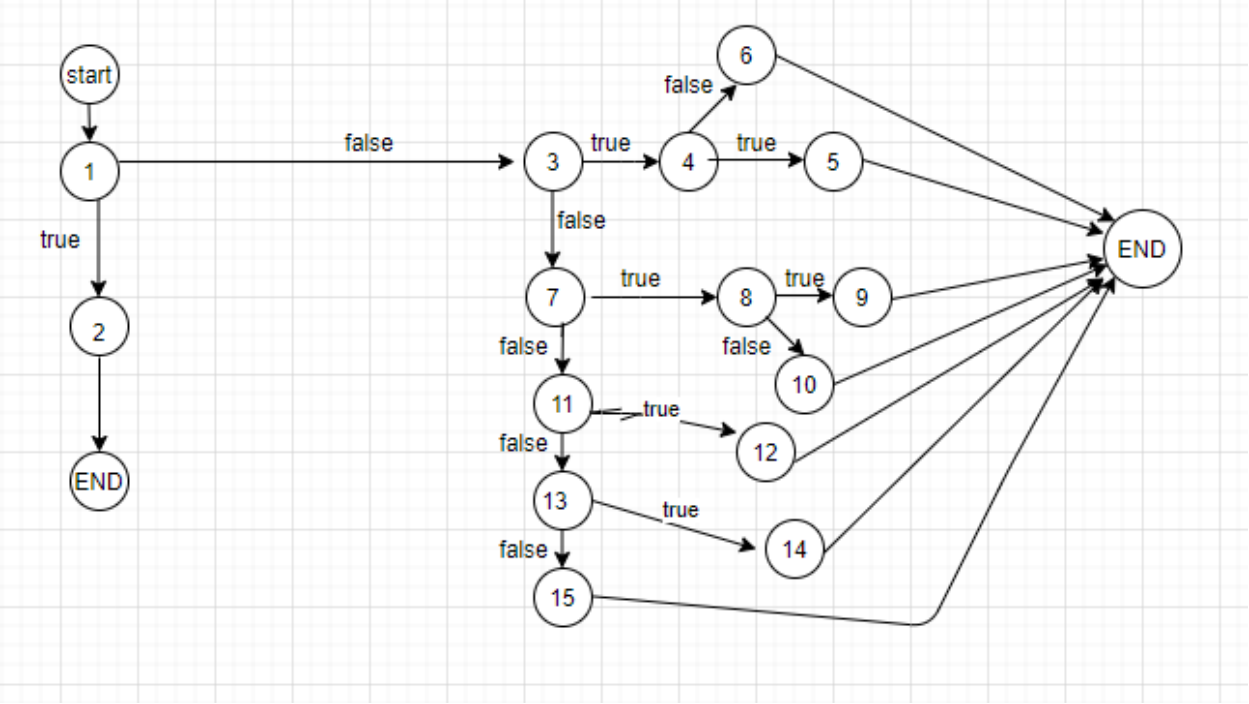


4. is_token_end

```
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 until meet another " */
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 until meet end of line */
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 tab or 59 */
193   return false;               /* other case,return FALSE */
194  }
```

Block	Lines	Entry	Exit
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
12	190b	190b	190b
13	191a	191a	191a
14	191b	191b	191b
15	193	193	193

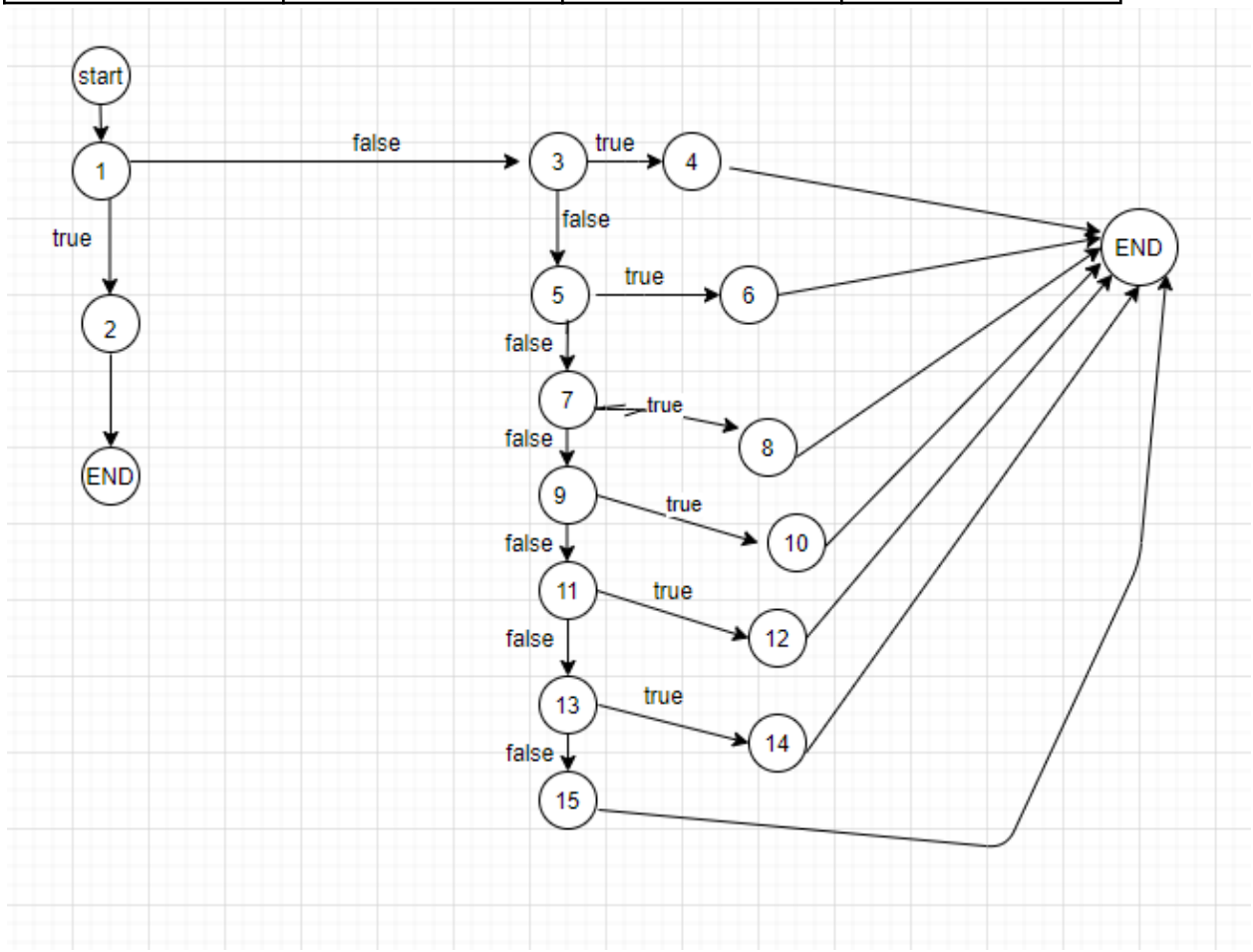


5. token_type

```
202     /*****************************************************/
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);                    /* else look as error token */
213     }
214
```

Block	Lines	Entry	Exit
1	205a	205a	205a
2	205b	205b	205b
3	206a	206a	206a
4	206b	206b	206b
5	207a	207a	207a
6	207b	207b	207b
7	208a	208a	208a
8	208b	208b	208b
9	209a	209a	209a
10	209b	209b	209b
11	210a	210a	210a

12	210b	210b	210b
13	211a	211a	211a
14	211b	211b	211b
15	212	212	212

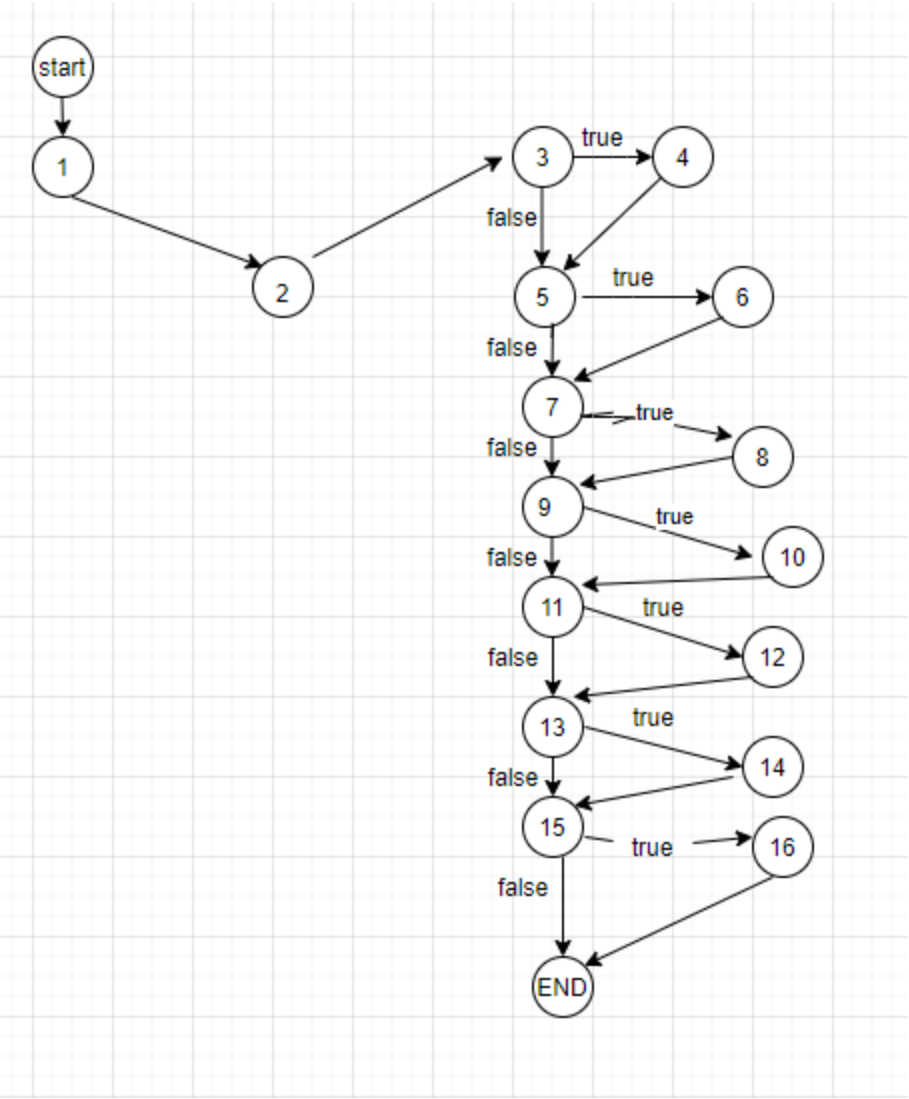


6. print_token

```
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   }
250
```

Block	Lines	Entry	Exit
1	220	220	220
2	221	221	221
3	222	222	222
4	224	224	224
5	227	227	227
6	229	229	229

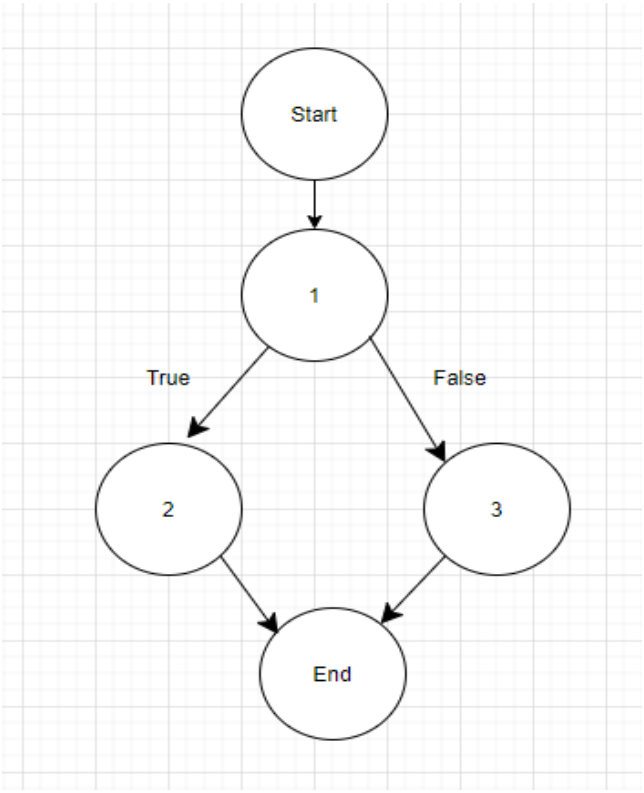
7	232a	232a	232a
8	232b	232b	232b
9	233	233	233
10	235	235	235
11	237	237	237
12	239	237	237
13	241	241	241
14	243	243	243
15	245	245	245
16	247	247	247



7. is_comment

```
259    static boolean is_comment(String ident)
260    {
261      if( ident.charAt(0) ==59 )   /* the char is 59   */
262         return true;
263      else
264         return false;
265    }
266
```

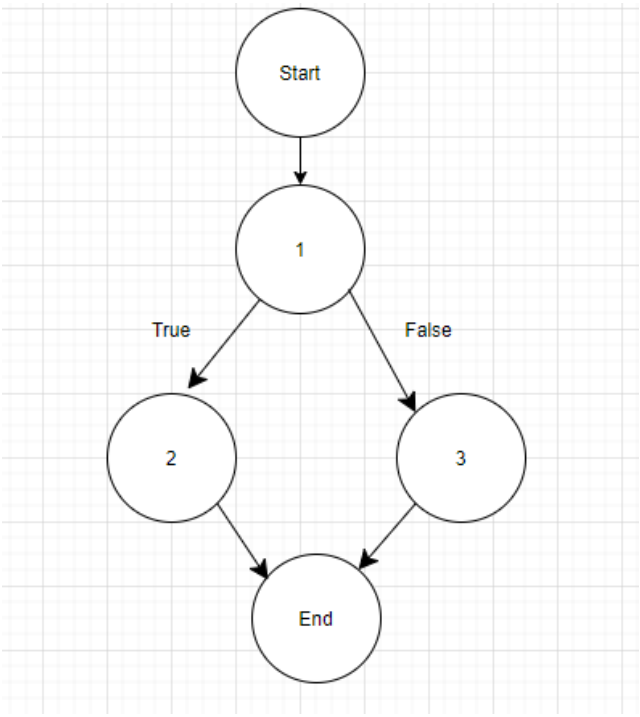
Block	Lines	Entry	Exit
1	261	261	261
2	262	262	262
3	264	264	264



8. 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    }
```

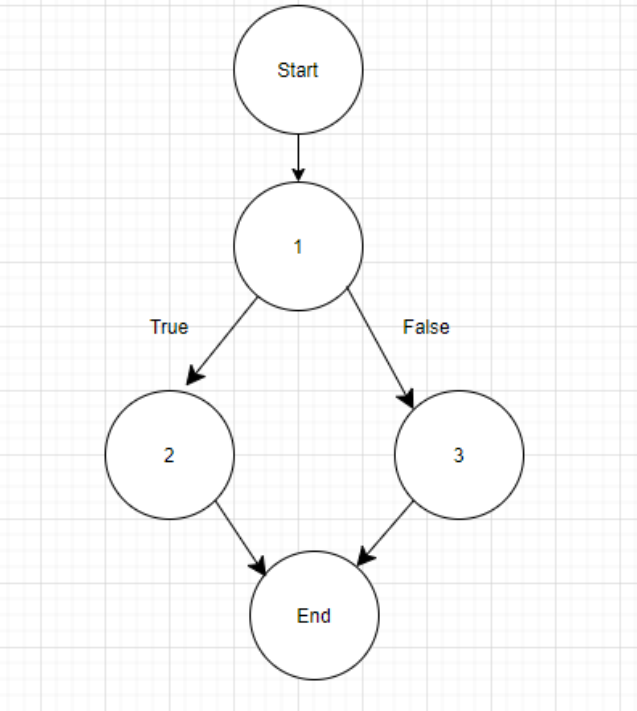
Block	Lines	Entry	Exit
1	288	288	288
2	289	289	289
3	291	291	291



9. 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    }
280
```

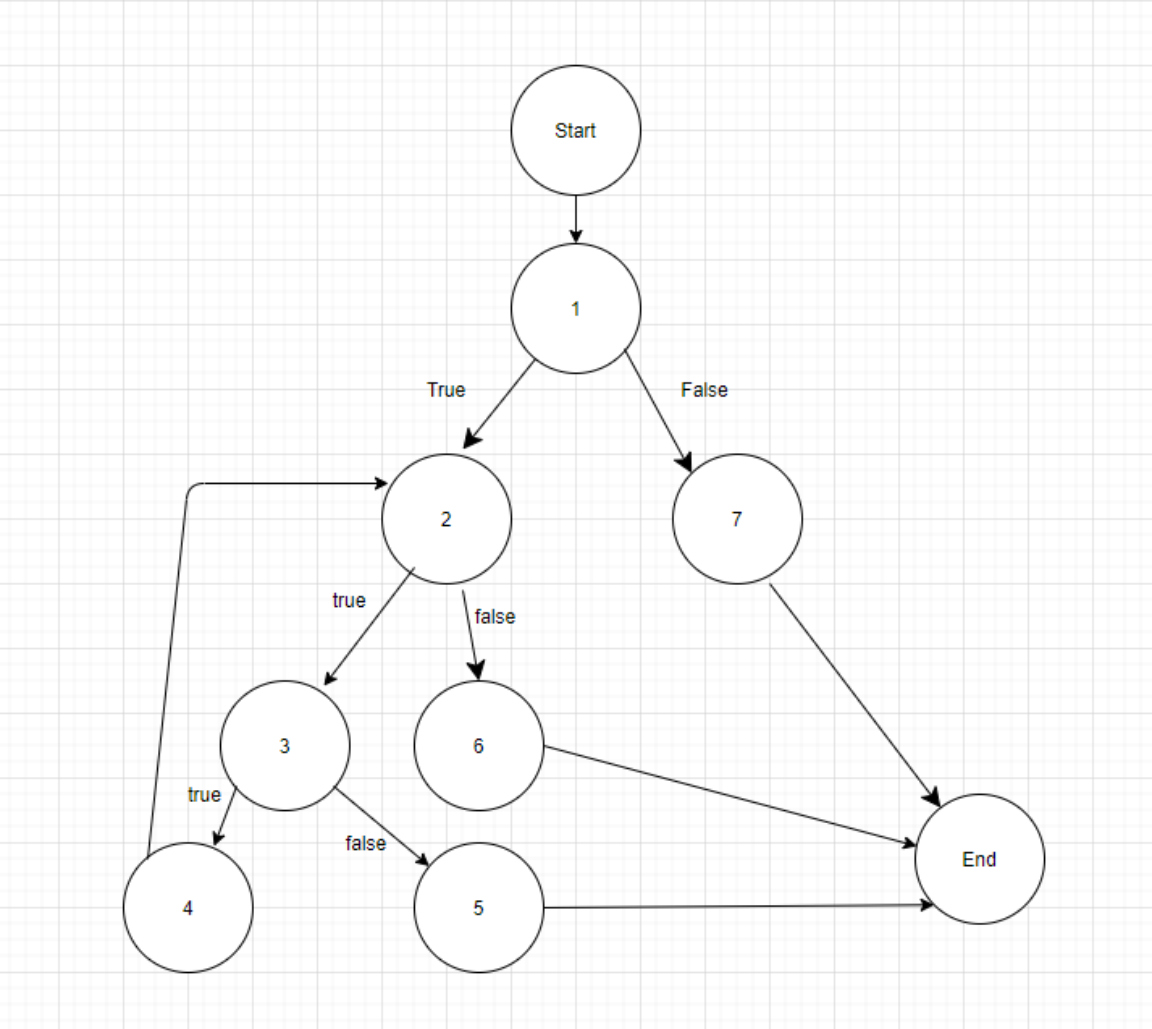
Block	Lines	Entry	Exit
1	274,275	274	275
2	276	276	276
3	278	278	278



10. is_num_constant

```
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' )   /* until meet token end sign */
306          {
307           if(Character.isDigit(str.charAt(i+1)))
308             i++;
309           else
310             return false;
311          }                          /* end WHILE */
312        return true;
313        }
314      else
315       return false;               /* other return FALSE */
316    }
```

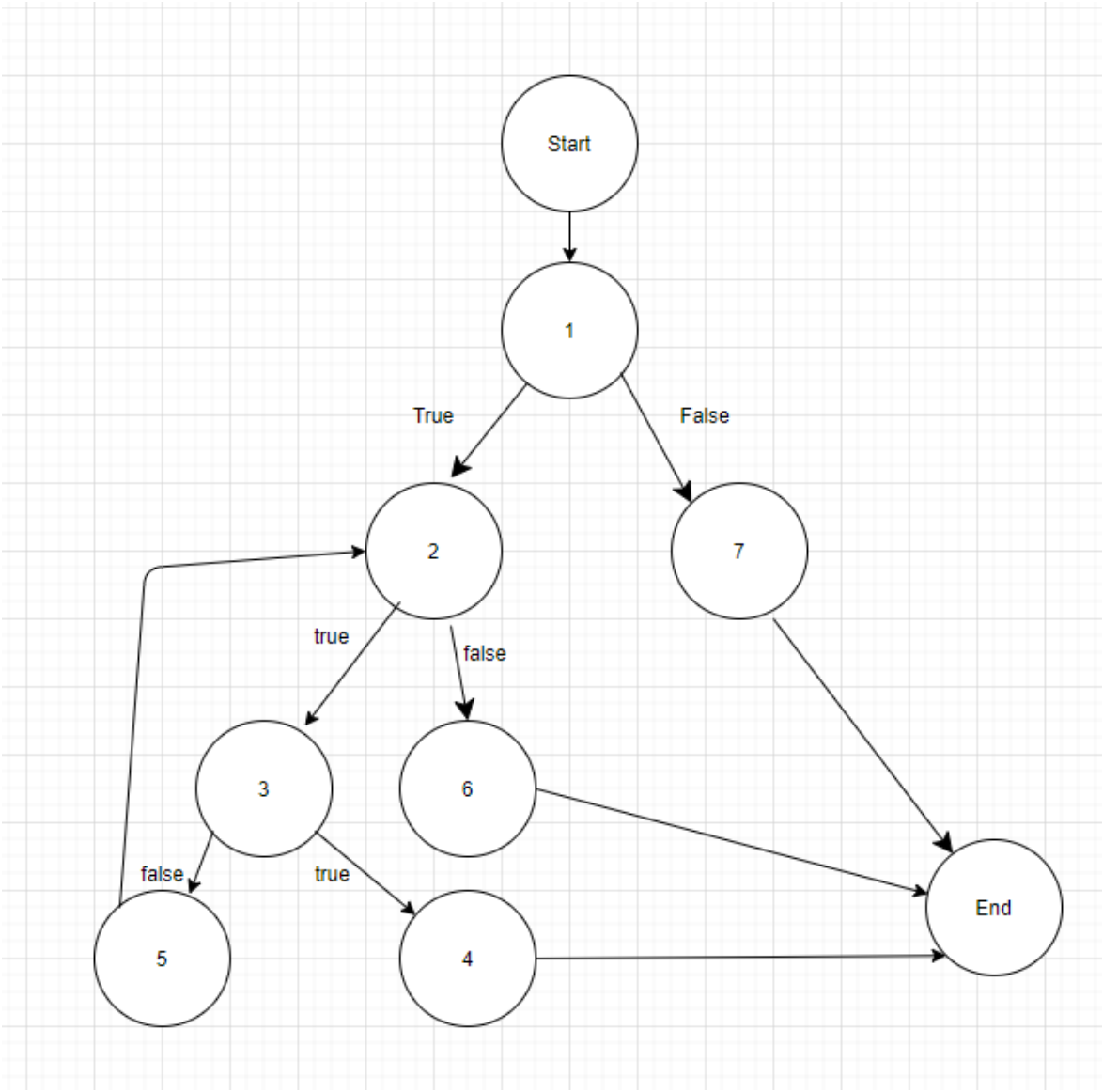
Block	Lines	Entry	Exit
1	301,303	301	303
2	305	305	305
3	307	307	307
4	308	308	308
5	310	310	310
6	312	312	312
7	315	315	315



11. is_str_constant

```
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(0)!='\0')
329             { if(str.charAt(i)=='"')
330                 return true;        /* meet the second '"'
331               else
332               i++;
333             }               /* end WHILE */
334        return true;
335        }
336      else
337        return false;       /* other return FALSE */
338    }
```

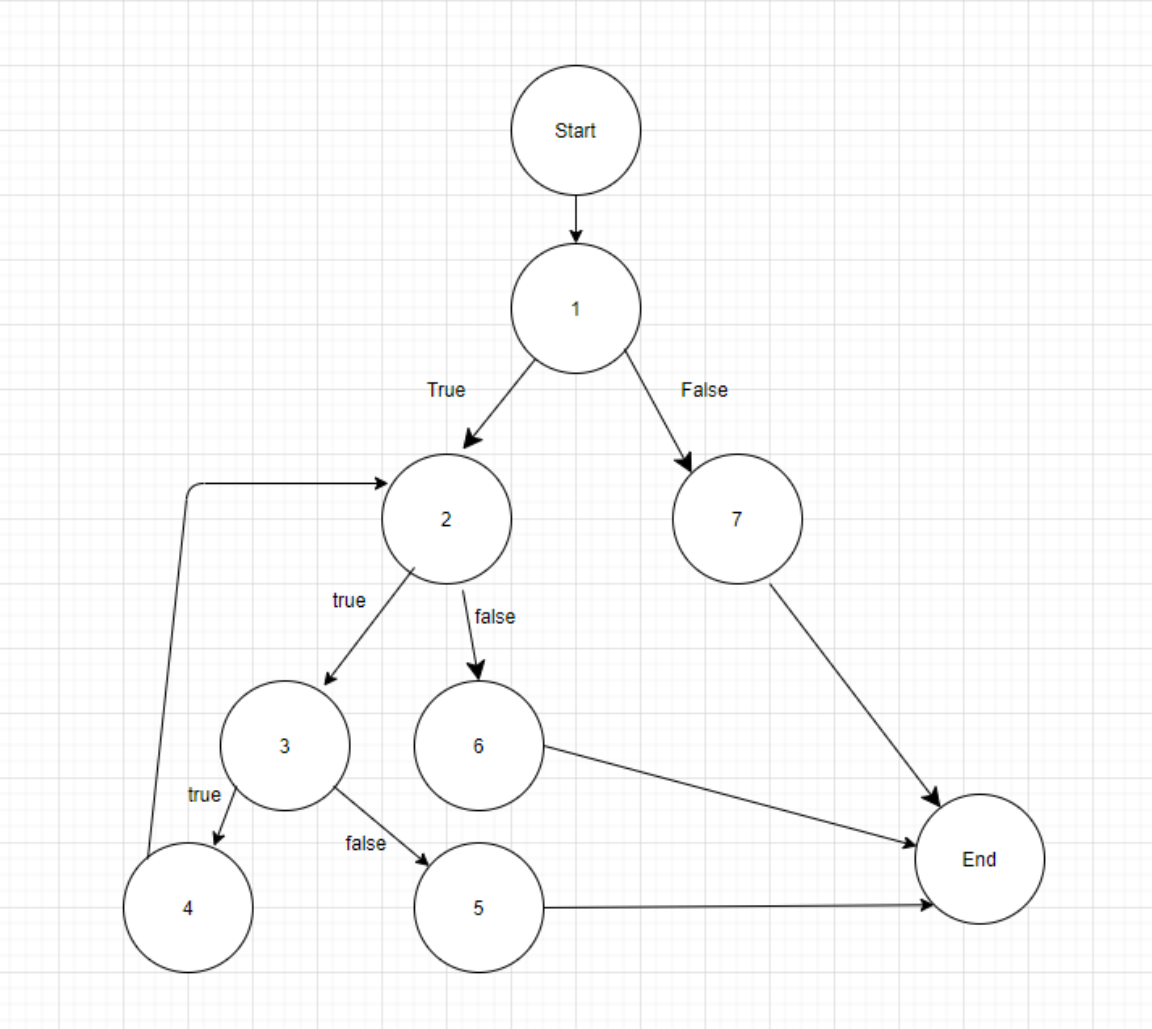
Block	Lines	Entry	Exit
1	325,327	325	327
2	328	328	328
3	329	329	329
4	330	330	330
5	332	332	332
6	334	334	334
7	337	337	337



12. 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' )   /* unti meet the end token sign */
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    }
363
364
```

Block	Lines	Entry	Exit
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

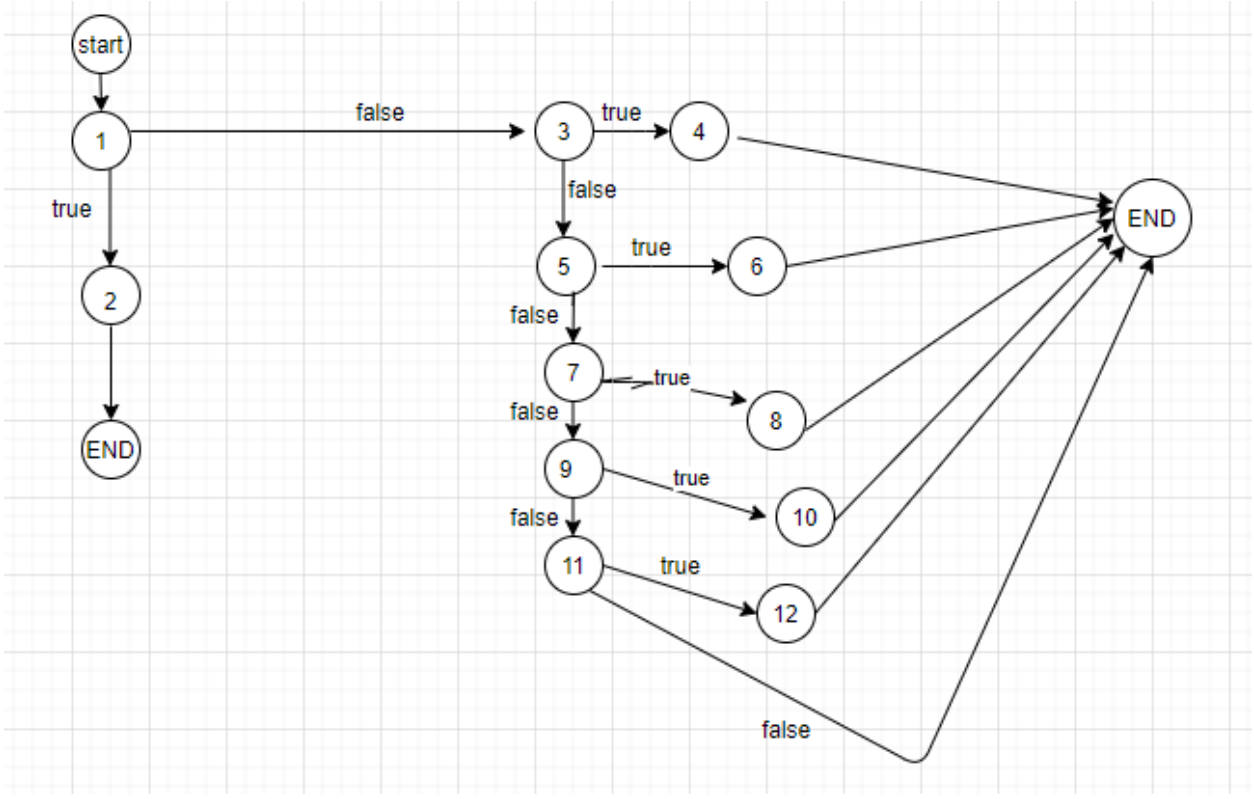


13. print_spec_symbol

```
372     static void print_spec_symbol(String str)
373     {
374         if      (str.equals("{"))
375         {
376
377                 System.out.print("lparen.\n");
378                 return;
379         }
380         if (str.equals(")"))
381         {
382
383                 System.out.print("rparen.\n");
384                 return;
385         }
386         if (str.equals("["))
387         {
388                 System.out.print("lsquare.\n");
389                 return;
390         }
391         if (str.equals("]"))
392         {
393
394                 System.out.print("rsquare.\n");
395                 return;
396         }
397         if (str.equals("'"))
398         {
399                 System.out.print("quote.\n");
400                 return;
401         }
402         if (str.equals("`"))
403         {
404
405                 System.out.print("bquote.\n");
406                 return;
407         }
408
409
410     }
```

Block	Lines	Entry	Exit
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

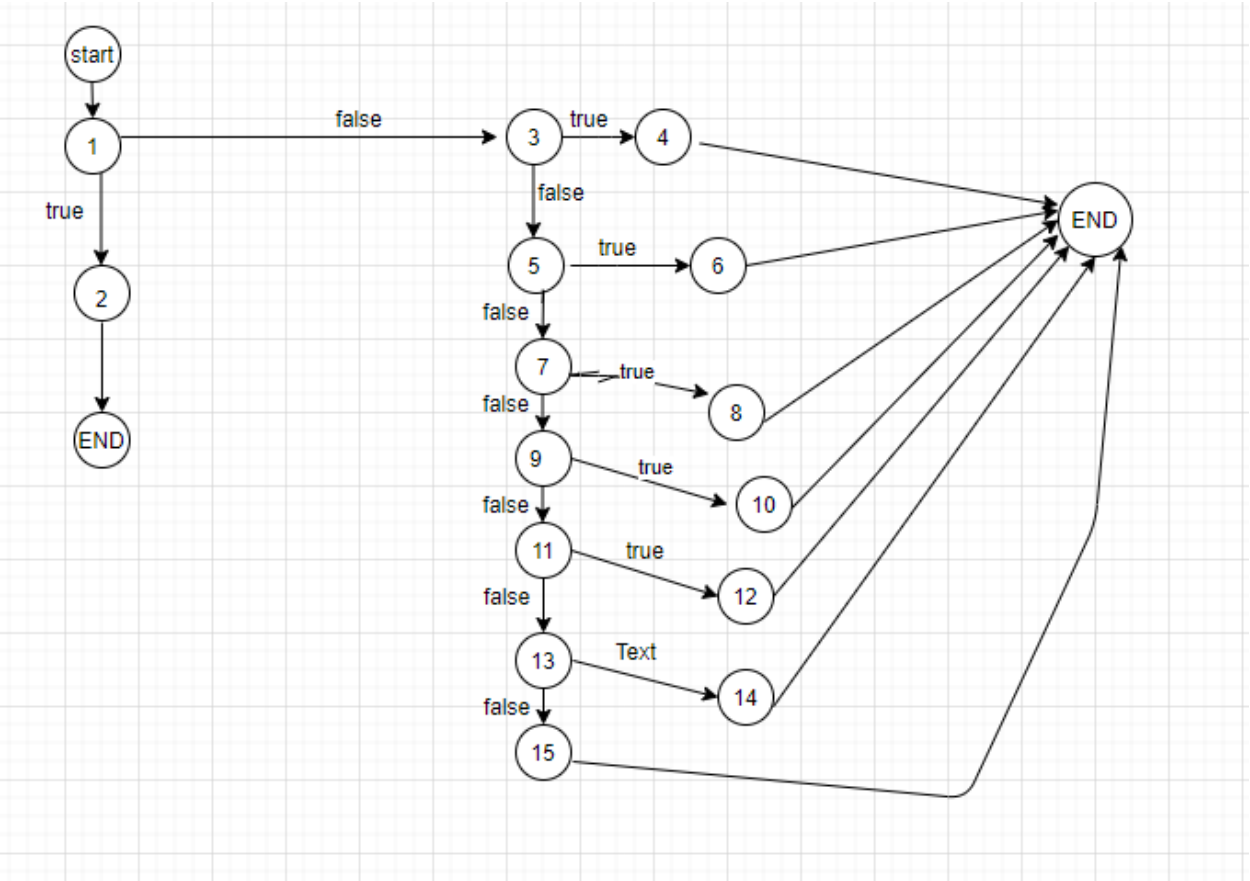


14. is_spec_symbol

```
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     }
```

Block	Lines	Entry	Exit
1	419	419	419
2	421	421	421
3	423	423	423
4	425	425	425
5	427	427	427

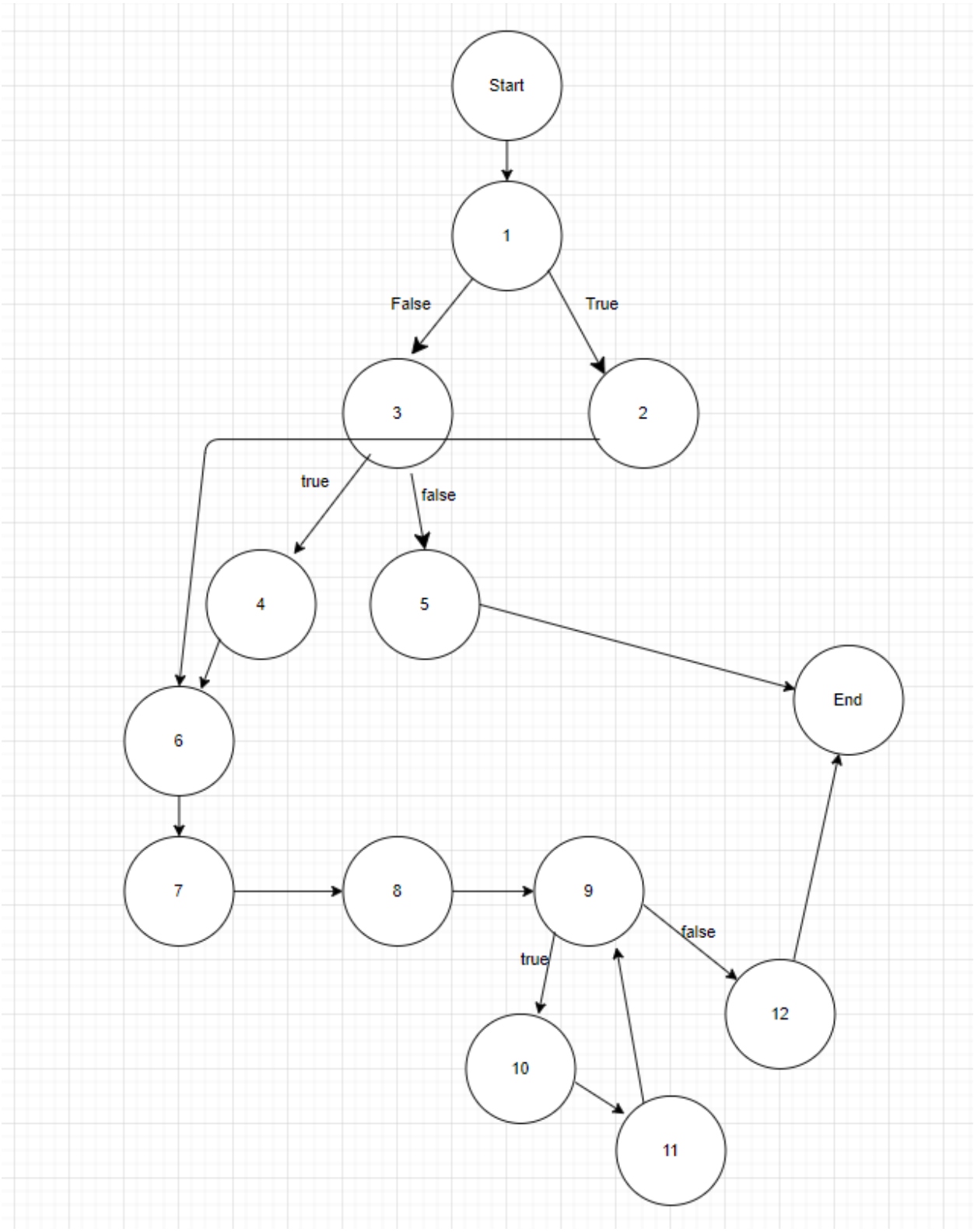
6	429	429	429
7	431	431	431
8	433	433	433
9	435	435	435
10	437	437	437
11	439	439	439
12	441	441	441
13	443	443	443
14	445	445	445
15	447	447	447



15. main

```
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 }
```

Block	Lines	Entry	Exit
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
8	463	463	463
9	464	464	464
10	465	465	465
11	466	466	466
12	470	470	470


Start
1
False
True
3
2
true
false
4
5
End
6
7
8
9
false
true
12
10
11