Nihar Gupte
1001556441
CFG Document - CFGs along with the corresponding source code, and the basic block table that identifies each basic block.

For the entirety of the document the following format shall be followed:
#. “name of the function in printtokens.java”
Source Code
Basic Block Table
Corresponding CFG diagram

For lines that have two statements, the first statement shall be labeled as ‘a’, the following as ‘b’ and so on.

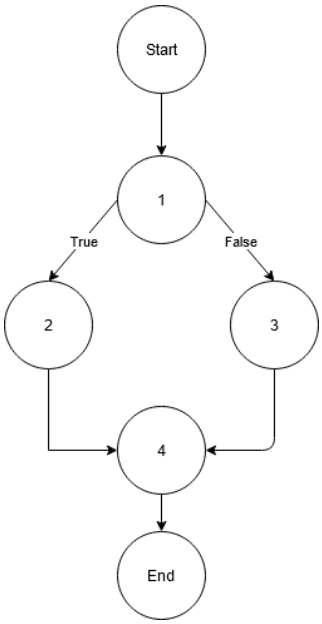
For the main method, the user-defined functions get their own separate basic block.

---

1. BufferedReader open_character_stream(String fname)

```
28    BufferedReader open_character_stream(String fname) {
29        BufferedReader br = null;
30        if (fname == null) {
31            br = new BufferedReader(new InputStreamReader(System.in));
32        } else {
33            try {
34                FileReader fr = new FileReader(fname);
35                br = new BufferedReader(fr);
36            } catch (FileNotFoundException e) {
37                System.out.print("The file " + fname +" doesn't exists\n");
38                e.printStackTrace();
39            }
40        }
41
42        return null;
43    }
```

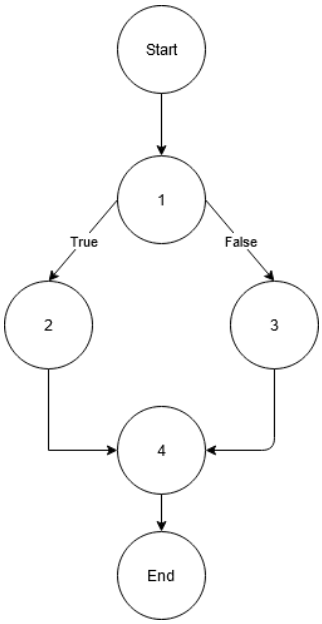
Block Number	Lines Number
1	29, 30
2	31
3	34, 35
4	42



2. BufferedReader open_token_stream(String fname)

```
83    BufferedReader open_token_stream(String fname)
84    {
85        BufferedReader br;
86     if(fname.equals(null))
87        br=open_character_stream(null);
88     else
89        br=open_character_stream(fname);
90     return br;
91    }
```

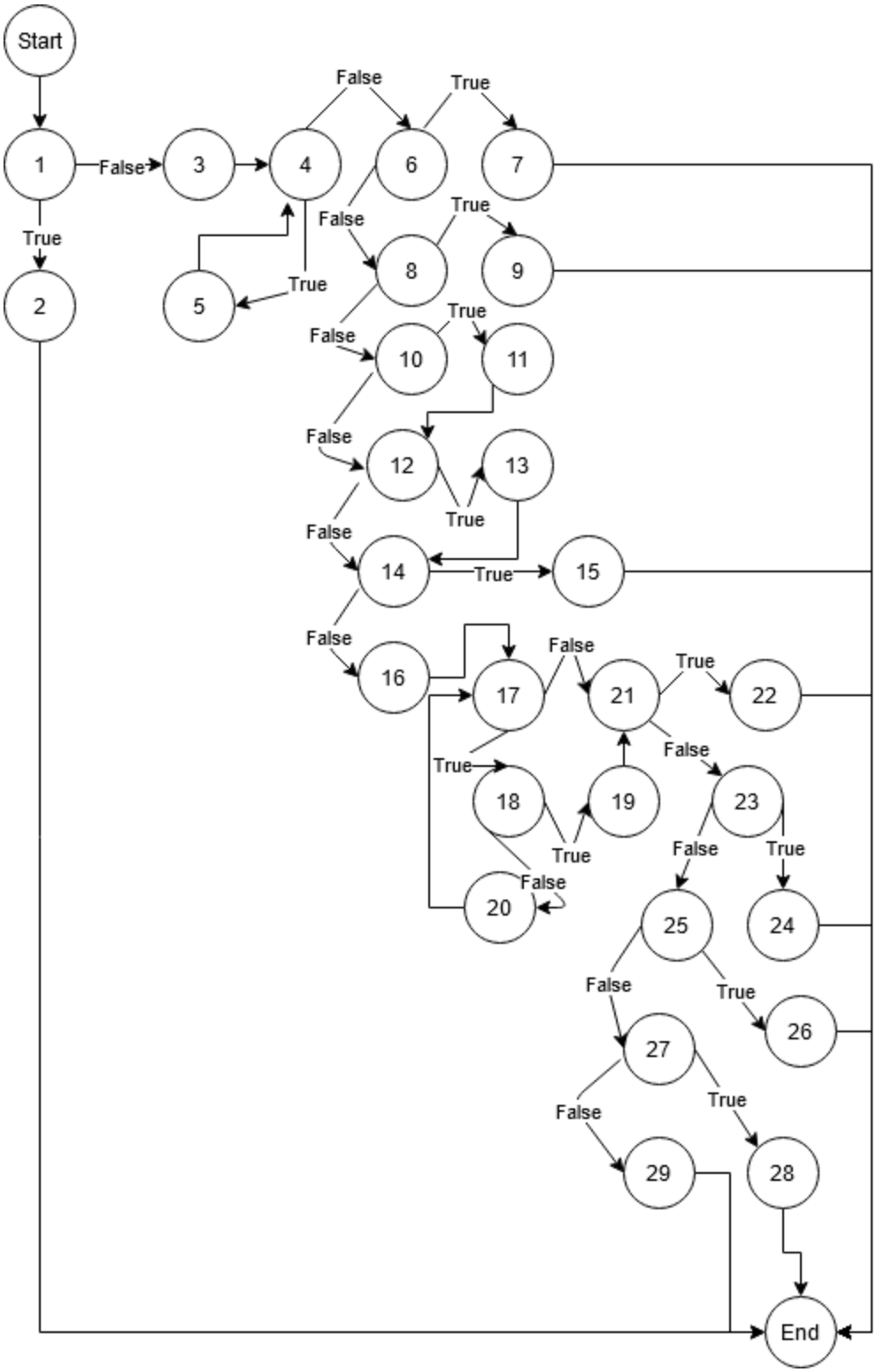
Block Number	Lines Number
1	85, 86
2	87
3	89
4	90



3. String get_token(BufferedReader br)

```
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')      /*
      {
        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 mee
   {
       while (is_token_end(id,res) == false)
       {
           sb.append(ch);
           br.mark(4);
           res = get_char(br);
           if (res == -1) {
               break;
           }
           ch = (char)res;
       }

       if(res == -1)       /* if end charact
          { unget_char(ch,br);        /* the
            return sb.toString();
          }

       if(is_spec_symbol(ch)==true)     /* i
          { unget_char(ch,br);        /* the
            return sb.toString();
          }
       if(id==1)                   /* if end
         {
           sb.append(ch);
           return sb.toString();
         }
       if(id==0 && ch==59)
                                          /* wh
         { unget_char(ch,br);       /* then
           return sb.toString();
         }
   } catch (IOException e) {
       e.printStackTrace();
   }

       return sb.toString();
   }
```

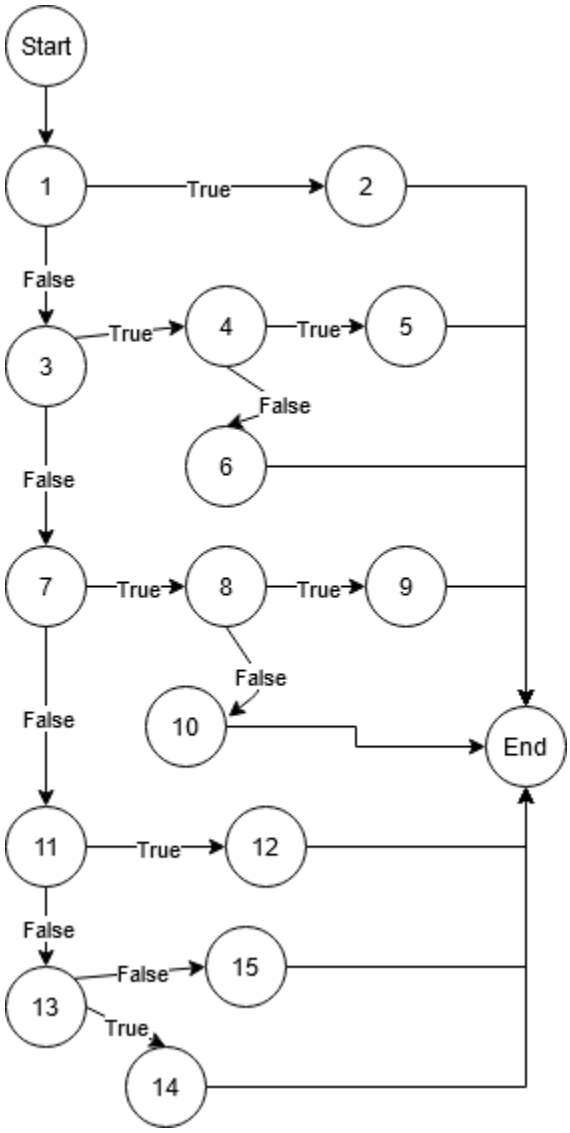
Block Number	Lines Number
1	102, 103, 104, 105, 107, 110, 111
2	112
3	114
4	115
5	117, 118
6	121a
7	121b
8	122, 123a
9	123b
10	124a
11	124b
12	125a
13	125b
14	127, 128
15	129, 130
16	132
17	134
18	136, 137, 138, 139
19	140
20	142
21	145
22	146, 147
23	150
24	151, 152
25	154
26	156, 157
27	159
28	161, 162
29	168


Start
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
End
False
True

4. boolean is_token_end(int str_com_id, int res)

```
176    static boolean is_token_end(int str_com_id, int res)
177    {
178     if(res==-1)return(true); /* is eof token? */
179     char ch = (char)res;
180     if(str_com_id==1)          /* is string token */
181        { if(ch=='"' | ch=='\n' || ch == '\r')   /* for string ur
182             return true;
183          else
184             return false;
185        }
186
187     if(str_com_id==2)    /* is comment token */
188       { if(ch=='\n' || ch == '\r' || ch=='\t')     /* for commer
189            return true;
190          else
191            return false;
192       }
193
194     if(is_spec_symbol(ch)==true) return true; /* is special_symb
195     if(ch ==' ' || ch=='\n'|| ch=='\r' || ch==59) return true;
196
197     return false;              /* other case,return FALSE */
198    }
```

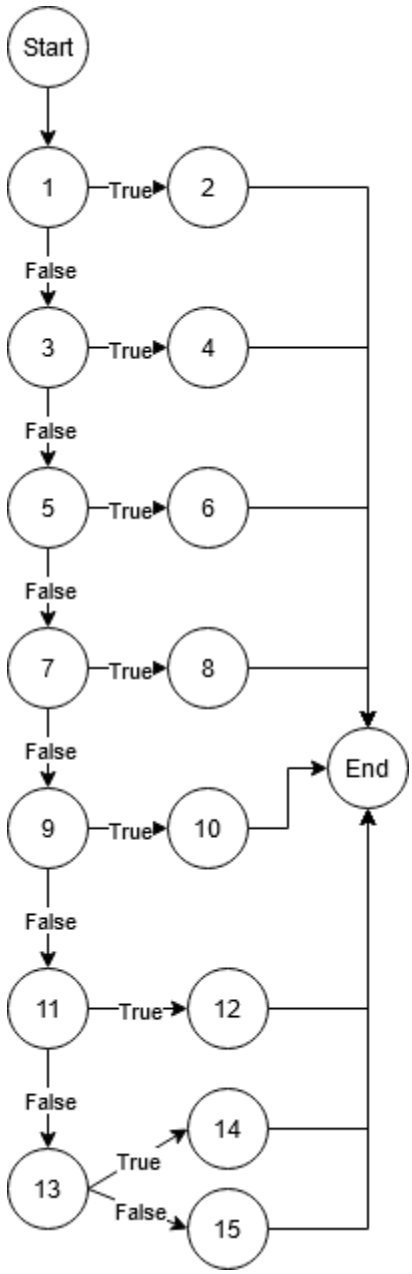
Block Number	Lines Number
1	178a
2	178b
3	179, 180
4	181
5	182
6	184
7	187
8	188
9	189
10	191
11	194a
12	194b
13	195a
14	195b
15	197



5. int token_type(String tok)

```
207    static int token_type(String tok)
208    {
209     if(is_keyword(tok))return(keyword);
210     if(is_spec_symbol(tok.charAt(0)))return(spec_symbol);
211     if(is_identifier(tok))return(identifier);
212     if(is_num_constant(tok))return(num_constant);
213     if(is_str_constant(tok))return(str_constant);
214     if(is_char_constant(tok))return(char_constant);
215     if(is_comment(tok))return(comment);
216     return(error);                    /* else look as erro
217    }
```

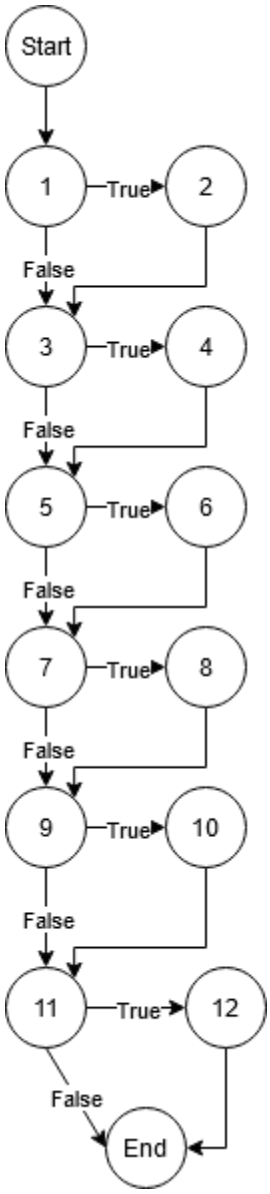
Block Number	Lines Number
1	209a
2	209b
3	210a
4	210b
5	211a
6	211b
7	212a
8	212b
9	213a
10	213b
11	214a
12	214b
13	215a
14	215b
15	216


Start
1
True
2
False
3
True
4
False
5
True
6
False
7
True
8
False
9
True
10
End
False
11
True
12
False
13
True
14
False
15

6. void print_token(String tok)

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

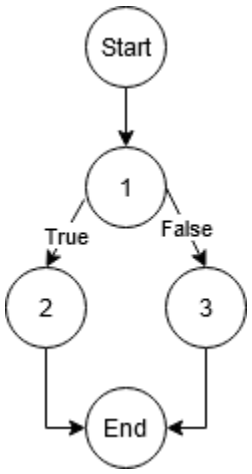
Block Number	Lines Number
1	224, 225, 226
2	228
3	231
4	233
5	236a
6	236b
7	237
8	239
9	241
10	243
11	245
12	247



7. boolean is_comment(String ident)

```
260    static boolean is_comment(String ident)
261    {
262      if( ident.charAt(0) ==59 )   /* the ch
263         return true;
264      else
265         return false;
266    }
267
```

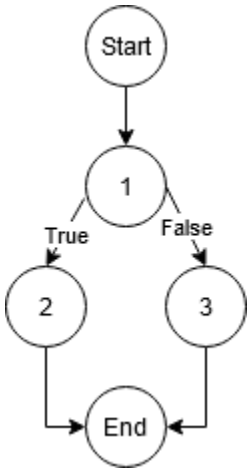
Block Number	Lines Number
1	262
2	263
3	265



8. boolean is_keyword(String str)

```
273     static boolean is_keyword(String str)
274     {
275      if (str.equals("and") || str.equals("or") || str.equals("if") ||
276              str.equals("xor")||str.equals("lambda")||str.equals("=>"))
277           return true;
278       else
279           return false;
280     }
```

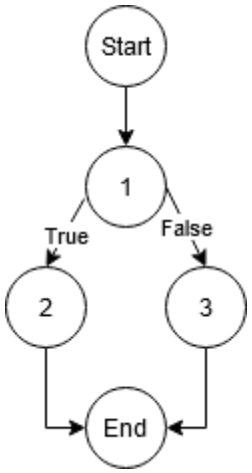
Block Number	Lines Number
1	275, 276
2	277
3	279



9. boolean is_char_constant(String str)

```
287     static boolean is_char_constant(String str)
288     {
289       if (str.length() > 2 && str.charAt(0)=='#' && Character.isLetter(str.charAt(1)))
290          return true;
291       else
292          return false;
293     }
```

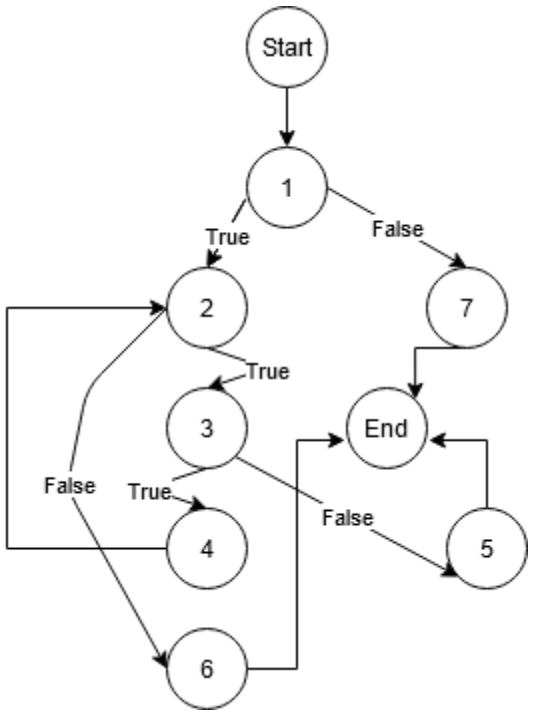
Block Number	Lines Number
1	289
2	290
3	292



10. boolean is_num_constant(String str)

```
300     static boolean is_num_constant(String str)
301     {
302       int i=1;
303
304       if ( Character.isDigit(str.charAt(0)))
305         {
306         while ( i <= str.length() && str.charAt(i) != '\0' )
307           {
308            if(Character.isDigit(str.charAt(i+1)))
309              i++;
310            else
311              return false;
312           }                              /* end WHILE */
313         return true;
314         }
315       else
316        return false;                    /* other return FALSE */
317     }
```

Block Number	Lines Number
1	302, 304
2	306
3	308
4	309
5	311
6	313
7	316

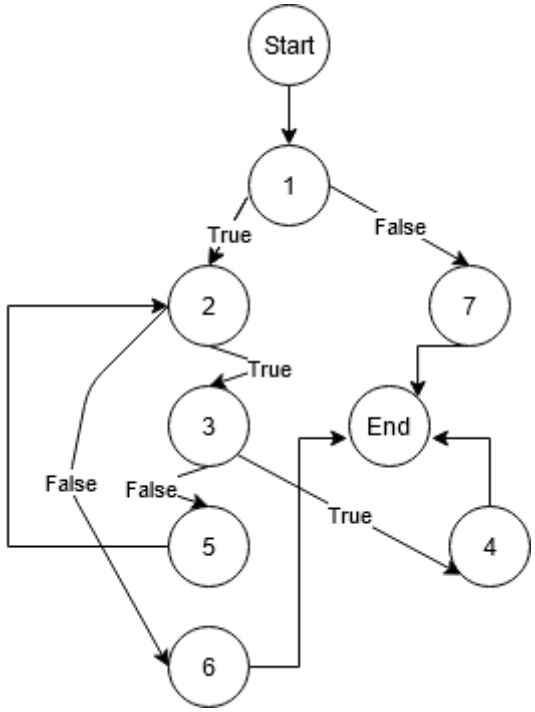


11. boolean is_str_constant(String str)

```
324    static boolean is_str_constant(String str)
325    {
326      int i=1;
327
328      if ( str.charAt(0) =='"')
329         { while (i < str.length() && str.charAt(0)!='\0')
330             { if(str.charAt(i)=='"')
331                 return true;        /* meet the second '"'
332               else
333               i++;
334             }                  /* end WHILE */
335         return true;
336        }
337      else
338        return false;       /* other return FALSE */
339    }
```

Block Number	Lines Number
1	326, 328
2	329
3	330

4	331
5	333
6	335
7	338

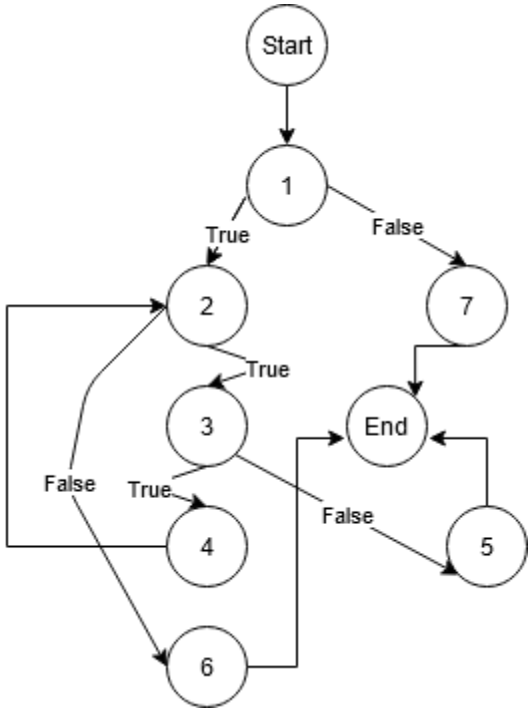


12. boolean is_identifier(String str)

```
346    static boolean is_identifier(String str)
347    {
348      int i=0;
349
350      if ( Character.isLetter(str.charAt(0)) )
351         {
352            while(i < str.length() && str.charAt(i) !='\0' )   /* unti meet the end token
353               {
354                if(Character.isLetter(str.charAt(i)) || Character.isDigit(str.charAt(i)))
355                   i++;
356                else
357                   return false;
358               }      /* end WHILE */
359         return false;
360         }
361      else
362         return true;
363    }
```

Block Number	Lines Number
1	348, 350
2	352
3	354

4	355
5	357
6	359
7	362

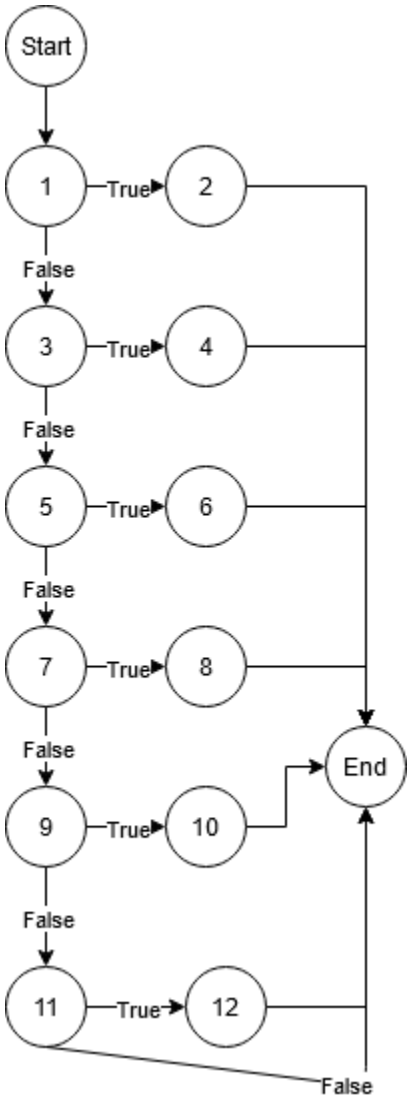


13. void print_spec_symbol(String str)

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

```
405        }
406        if (str.equals("'"))
407        {
408                System.out.print("quote.\n");
409                return;
410        }
411        if (str.equals("`"))
412        {
413
414                System.out.print("bquote.\n");
415                return;
416        }
417
418
419    }
420
```

Block Number	Lines Number
1	383
2	386, 387
3	389
4	392, 393
5	395
6	397, 398
7	400
8	403, 404
9	406
10	408, 409
11	411
12	414, 415

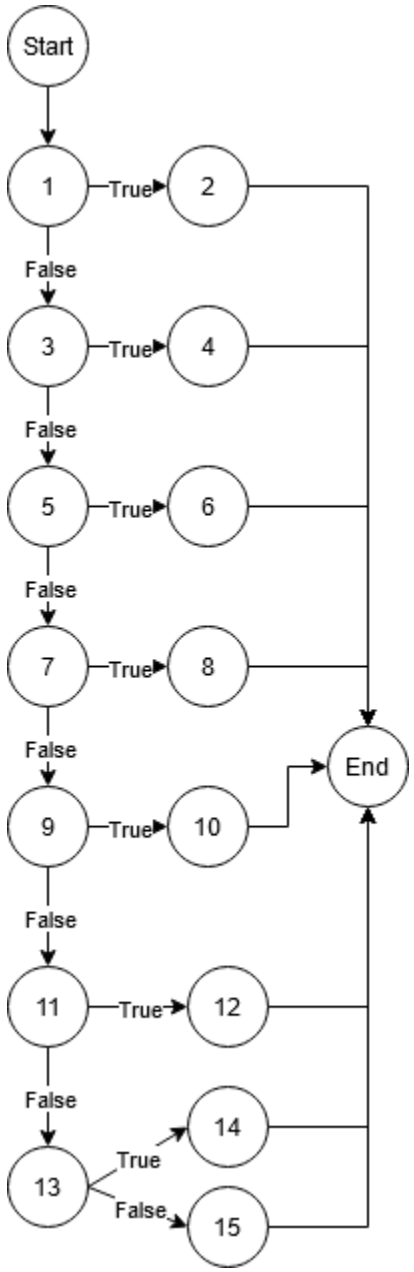


14. boolean is_spec_symbol(char c)

```
426     static boolean is_spec_symbol(char c)
427     {
428         if (c == '(')
429         {
430             return true;
431         }
432         if (c == ')')
433         {
434             return true;
435         }
436         if (c == '[')
437         {
438             return true;
439         }
440         if (c == ']')
441         {
442             return true;
443         }
444         if (c == '/')
445         {
446             return true;
447         }
448         if (c == '`')
449         {
450             return true;
451         }
452         if (c == ',')
453         {
454             return true;
455         }
456         return false;     /* others return
457     }
458
```

Block Number	Lines Number
1	428
2	430
3	432
4	434
5	436

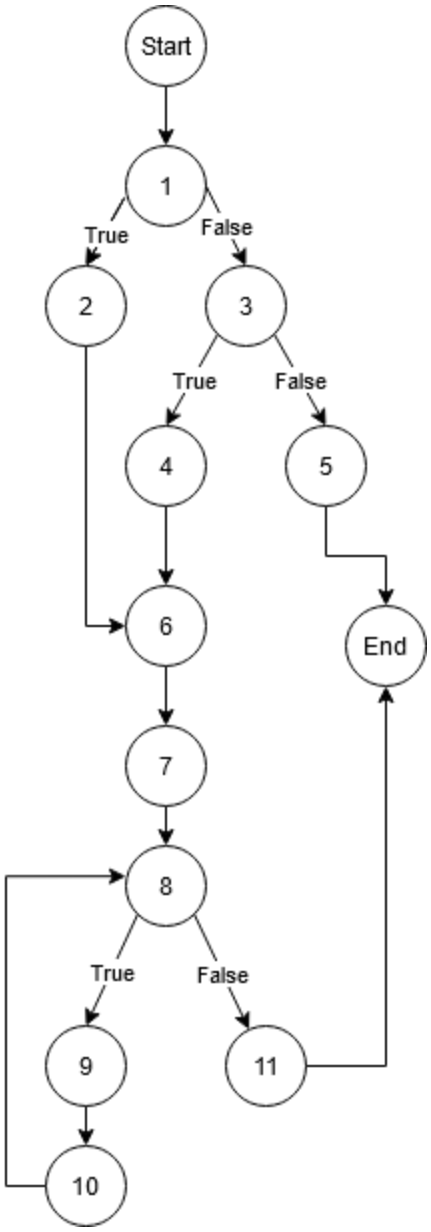
6	438
7	440
8	442
9	444
10	446
11	448
12	450
13	452
14	454
15	456



15. void main(String[] args)

```
459     public static void main(String[] args) throws IOException {
460         String fname = null;
461         if (args.length == 0) { /* if not given filename,take as '""' *
462             fname = new String();
463         } else if (args.length == 1) {
464             fname = args[1];
465         } else {
466             System.out.print("Error!,please give the token stream\n");
467             System.exit(0);
468         }
469         Printtokens t = new Printtokens();
470         BufferedReader br = t.open_token_stream(fname); /* open token s
471         String tok = t.get_token(br);
472         while (tok != null) {   /* take one token each time until eof *
473             t.print_token(tok);
474             tok = t.get_token(br);
475         }
476 
477         System.exit(0);
478     }
479 }
```

Block Number	Lines Number
1	460, 461
2	462
3	463
4	464
5	466, 467
6	469, 470
7	471
8	472
9	473
10	474
11	477


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