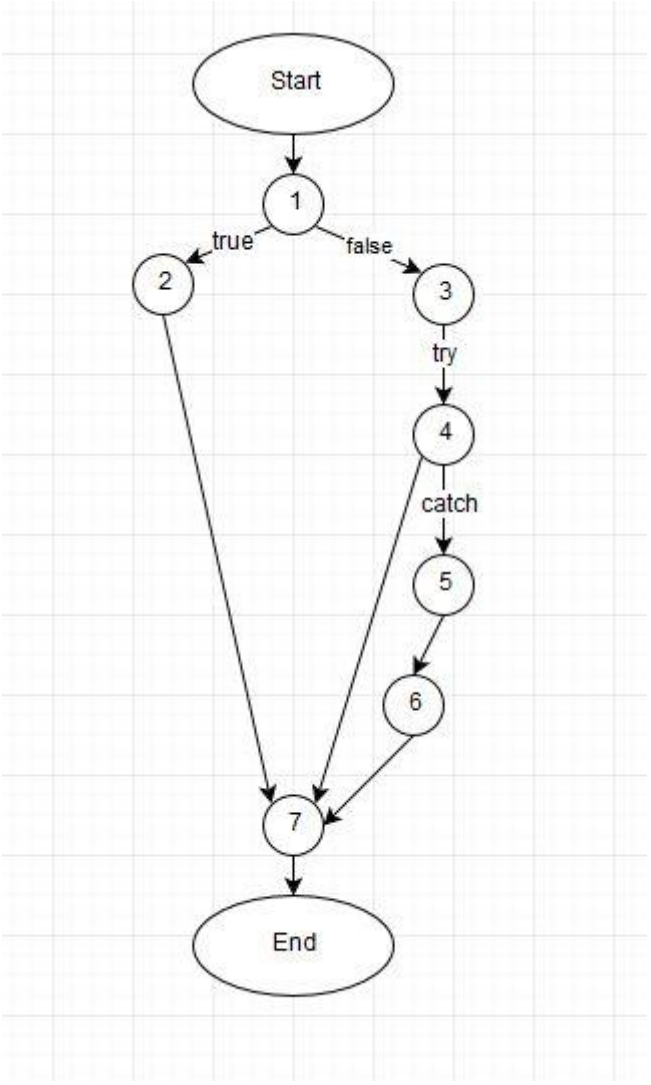
CFG for each methods and their corresponding basic block

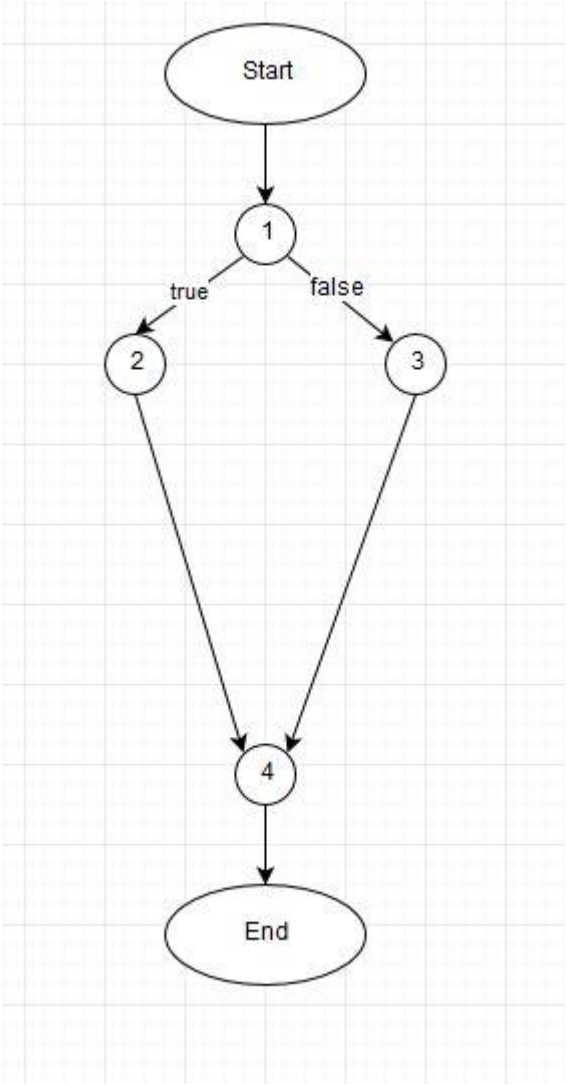
Function Name: open_character_stream



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	6	6	6
4	7,8	7	8
5	10	10	10
6	11	11	11
7	12	12	12

```
1       begin
2       BufferedReader br = null;
3       if (fname == null) {
4          br = new BufferedReader(new InputStreamReader(System.in));
5       } else {
6          try {
7             FileReader fr = new FileReader(fname);
8             br = new BufferedReader(fr);
9          } catch (FileNotFoundException e) {
10            System.out.print("The file " + fname + " doesn't exists\n");
11            e.printStackTrace();
12       return br;
13       end
```

Function Name: open_token_stream

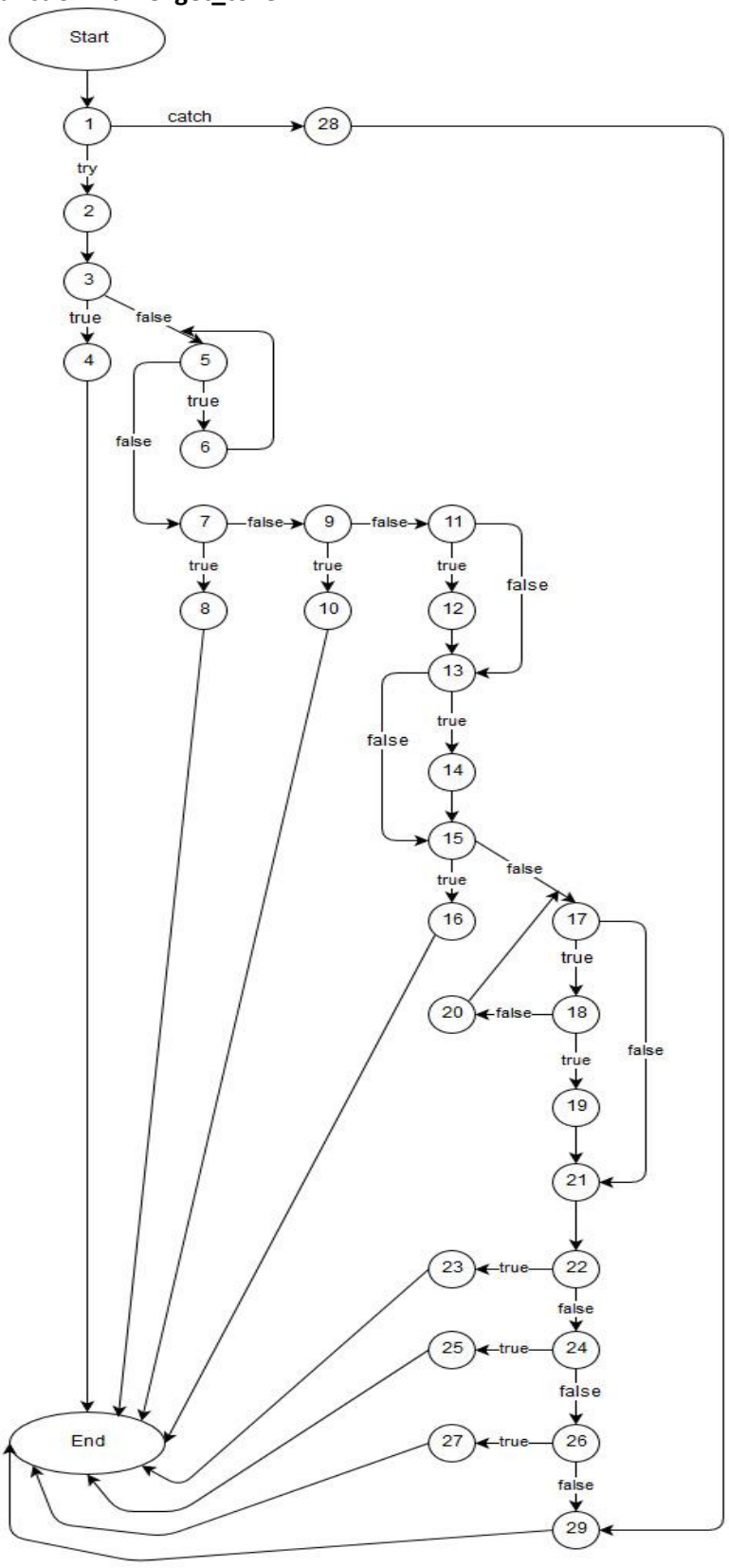


Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	6	6	6
4	7	7	7

```
1. begin
2.    BufferedReader br;
3.    if(fname.equals(null))
4.        br=open_character_stream(null);
5.    else
6.      br=open_character_stream(fname);
7.    return br;
8. end
```

Function Name: get_token

Block	Lines	Entry	Exit
1	2,3,4,5,6,7	2	7
2	8	8	8

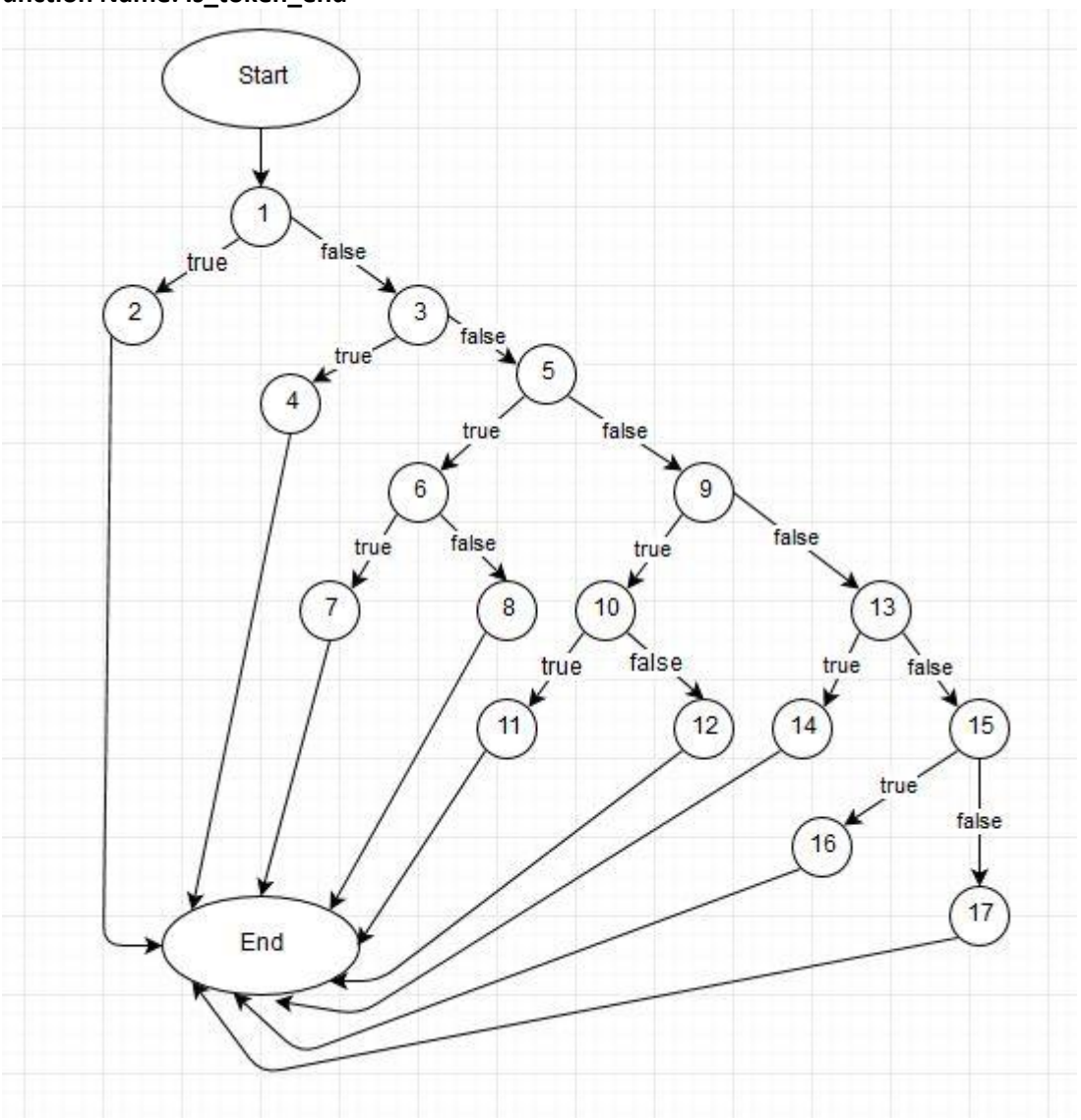


3	9	9	9
4	10	10	10
5	11,12	11	12
6	13,14	13	14
7	15	15	15
8	16	16	16
9	17,18	17	18
10	19	19	19
11	20	20	20
12	21	21	21
13	22	22	22
14	23	23	23
15	24,25	24	25
16	26,27	26	27
17	28,29	28	29
18	30,31, 32,33	30	33
19	34	34	34
20	35	35	36
21	36,37, 38	37	38
22	39	39	39
23	40,41	40	41
24	42	42	42
25	43,44	43	44
26	45	45	45
27	46,47	46	47
28	49	49	49
29	50	50	50

```
1   begin
2       int i = 0, j;
3       int id = 0;
4       int res = 0;
5       char ch = '\0';
6       StringBuilder sb = new StringBuilder();
7       try {
8          res = get_char(br);
9          if (res == -1)
10             return null;
11        ch = (char) res;
12         while (ch == ' ' || ch == '\n' || ch == '\r')      /* strip all blanks until meet characters */ {
13            res = get_char(br);
14            ch = (char) res;
15         if (res == -1)
16           return null;
17         sb.append(ch);
18         if (is_spec_symbol(ch))
19           return sb.toString();
20         if (ch == '"')
21           id = 2;    /* prepare for string */
22         if (ch == 59)
23           id = 1;    /* prepare for comment */
24         res = get_char(br);
25         if (res == -1)
```

```
26            unget_char(ch, br);
27            return sb.toString();
28         ch = (char) res;
29         while (!is_token_end(id, res))/* until meet the end character */
30            sb.append(ch);
31            br.mark(4);
32            res = get_char(br);
33            if (res == -1)
34               break;
35            ch = (char) res;
36         if (res == -1)      /* if end character is eof token    */ {
37            unget_char(ch, br);       /* then put back eof on token_stream */
38            return sb.toString();
39         if (is_spec_symbol(ch))    /* if end character is special_symbol */ {
40            unget_char(ch, br);       /* then put back this character      */
41            return sb.toString();
42         if (id == 1)               /* if end character is '' and is string */ {
43            sb.append(ch);
44            return sb.toString();
45         if (id == 0 && ch == 59)
46            unget_char(ch, br);      /* then put back this character        */
47            return sb.toString();
48       catch (IOException e)
49          e.printStackTrace();
50       return sb.toString();               /* return normal case token           */
51   end
```

Function Name: is_token_end

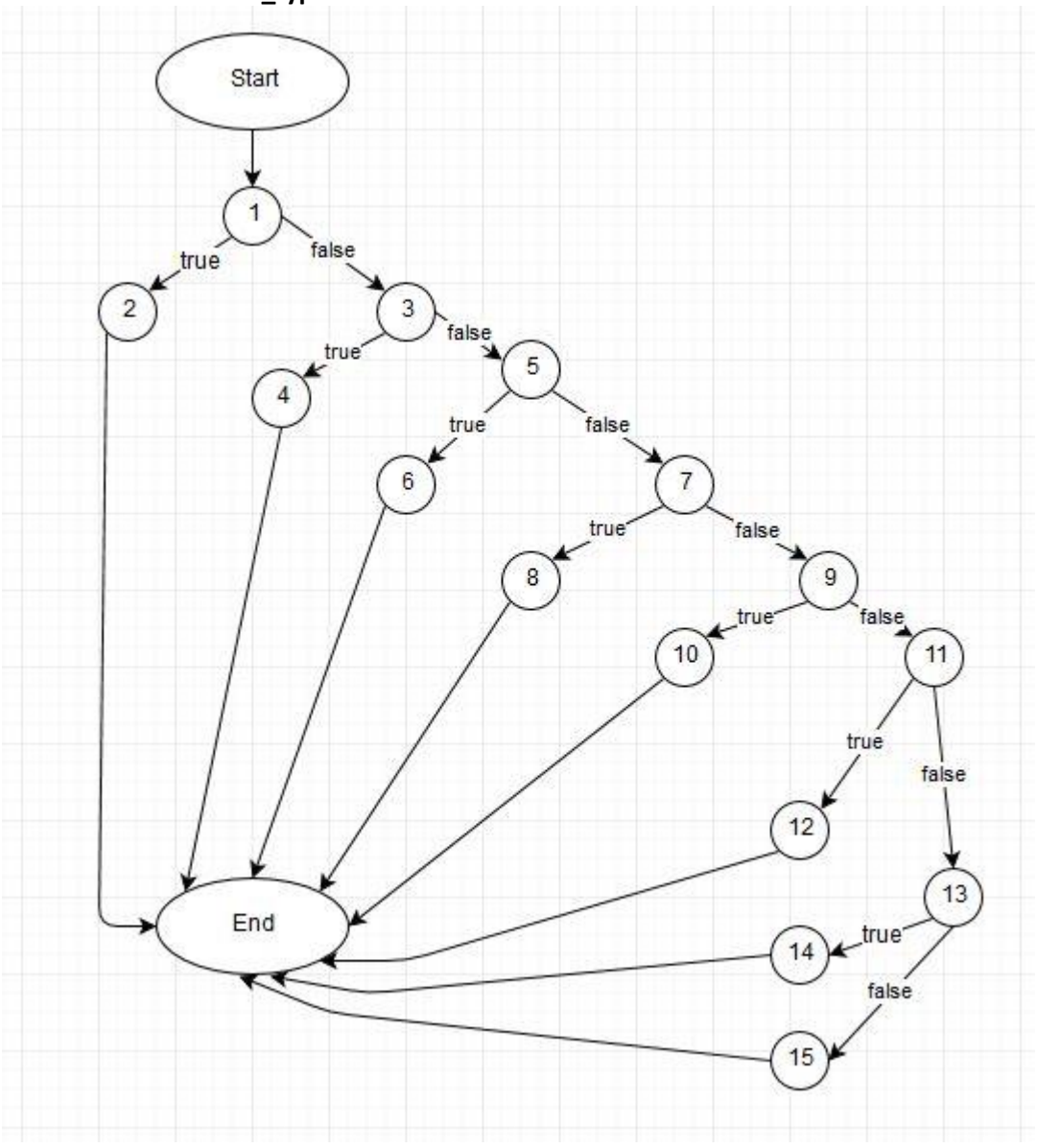


Block	Lines	Entry	Exit
1	2	2	2
2	3	3	3
3	4,5	4	5
4	6	6	6
5	7	7	7
6	8	8	8
7	9	9	9
8	11	11	11
9	12	12	12
10	13	13	13
11	14	14	14
12	16	16	16
13	17	17	17
14	18	18	18
15	19	19	19
16	20	20	20
17	21	21	21

```
1 begin
2       if (res == -1)
3               return (true);
4       char ch = (char) res;
```

```
5     if (ch == '"' | ch == '\n' || ch == '\r ')
6        return true;
7     if (str_com_id == 1)
8           if (ch == '\n' || ch == '\r' || ch == ' ' || ch == 59)
9                 return true;
10          else
11                return false;
12    if (str_com_id == 2)
13         if (ch == '\n' || ch == '\r' || ch == '\t ')
14                return true;
15          else
16                return false;
17    If (is_spec_symbol(ch) == true)
18          return true;
19    If (ch == ' ' || ch == '\n' || ch == '\r' || ch == 59)
20          return true;
21    return false
22    end
```

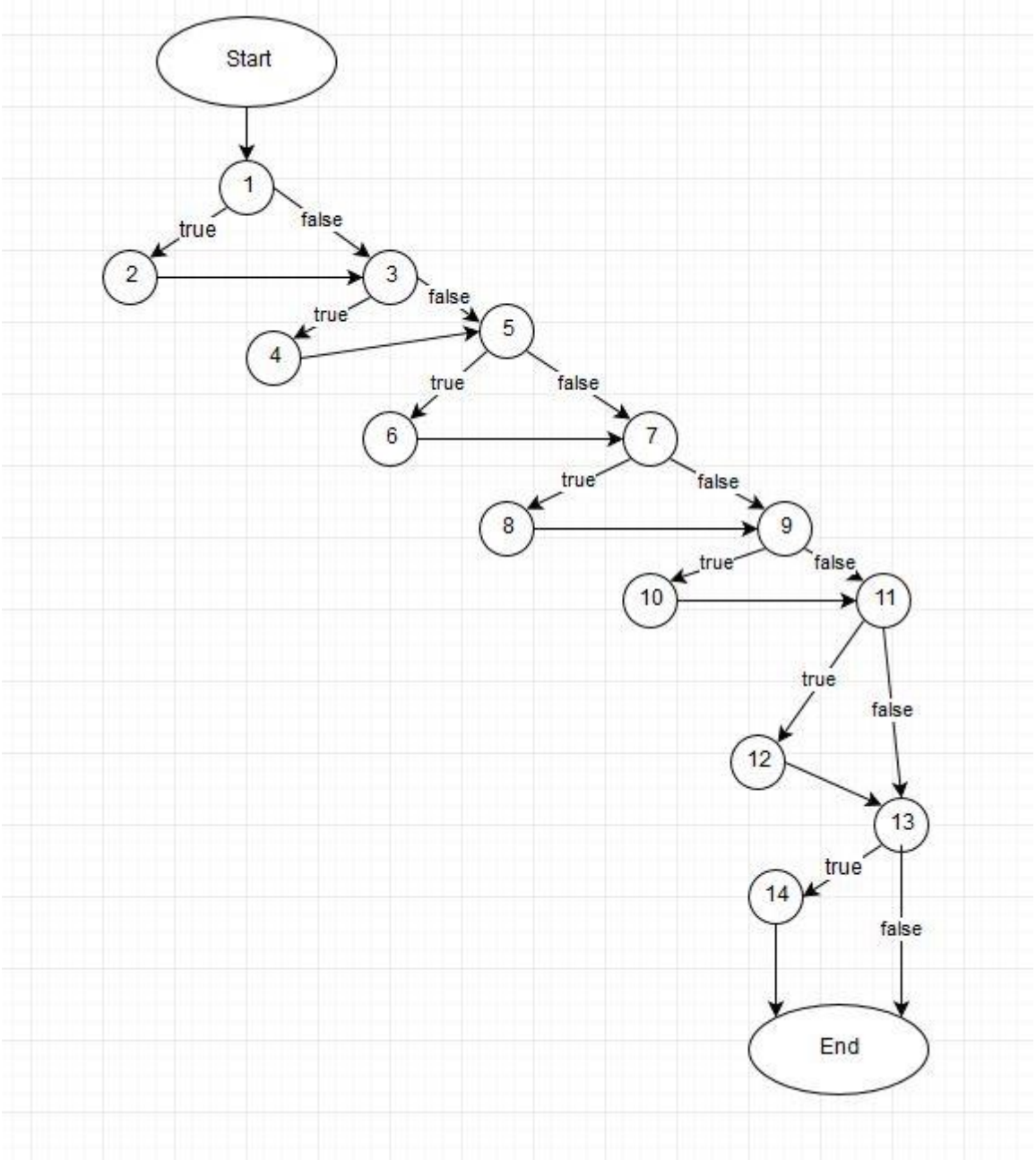
Function Name: token_type



Block	Lines	Entry	Exit
1	2	2	2
2	3	3	3
3	4	4	4
4	5	5	5
5	6	6	6
6	7	7	7
7	8	8	8
8	9	9	9
9	10	10	10
10	11	11	11
11	12	12	12
12	13	13	13
13	14	14	14
14	15	15	15
15	16	16	16

```
1    begin
2       if (is_keyword(tok))
3            return (keyword);
4       if (is_spec_symbol(tok.charAt(0)))
5            return (spec_symbol);
6       if (is_identifier(tok))
7            return (identifier);
8       if (is_num_constant(tok))
9            return (num_constant);
10      if (is_str_constant(tok))
11           return (str_constant);
12      if (is_char_constant(tok))
13           return (char_constant);
14      if (is_comment(tok))
15           return (comment);
16      return (error);
17   end
```

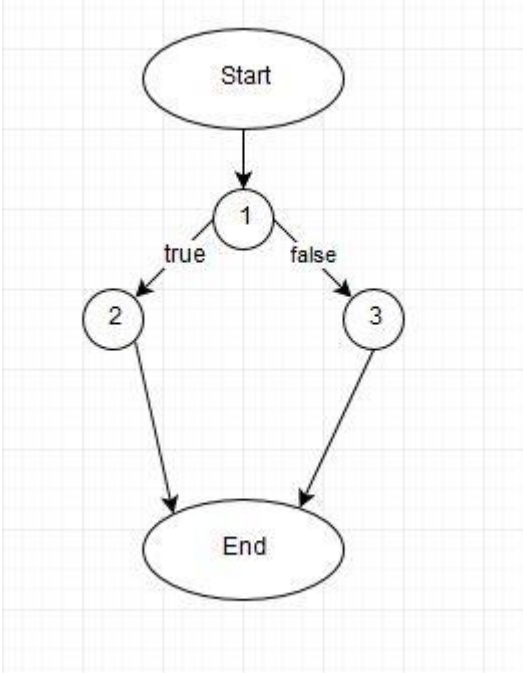
Function Name: print_token



Block	Lines	Entry	Exit
1	2,3,4	2	4
2	5	5	5
3	6	6	6
4	7	7	7
5	8	8	8
6	9	9	9
7	10	10	10
8	11	11	11
9	12	12	12
10	13	13	13
11	14	14	14
12	15	15	15
13	16	16	16
14	17	17	17

```
1 begin
2       int type;
3       type = token_type(tok);
4       if (type== error)
5             System.out.print("error,\"" + tok + "\".\n");
6      if (type== keyword)
7             System.out.print("keyword,\"" + tok + "\".\n");
8      if (type== spec_symbol)
9             print_spec_symbol(tok);
10     if (type== identifier)
11            System.out.print("identifier,\"" + tok + "\".\n");
12     if (type== num_constant)
13            System.out.print("numeric," + tok + ".\n");
14     if (type== str_constant)
15            System.out.print("string," + tok + ".\n");
16     if (type== char_constant)
17            System.out.print("character,\"" + tok.charAt(1) + "\".\n");
18  end
```

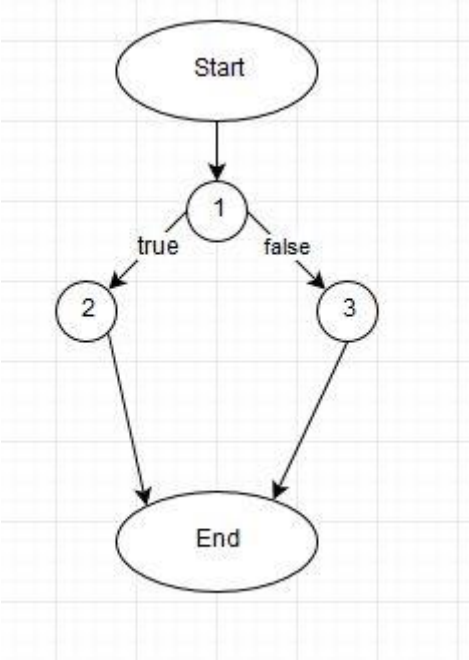
Function Name: is_comment



Block	Lines	Entry	Exit
1	2	2	2
2	3	3	3
3	5	5	5

```
1   begin
2       if (ident.charAt(0) == 59)
3           return true;
4       else
5           return false;
6   end
```

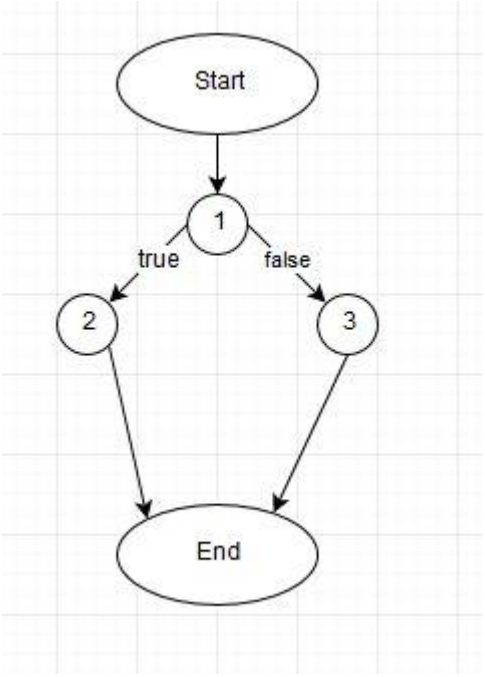
Function Name: is_keyword



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	6	6	6

```
1 begin
2       if ( str.equals("and") ||str.equals("or") ||str.equals("if") ||
3 str.equals("xor") ||str.equals("lambda") ||str.equals("=>"))
4       return true;
5 else
6       return false;
7 end
```

Function Name: is_char_constant



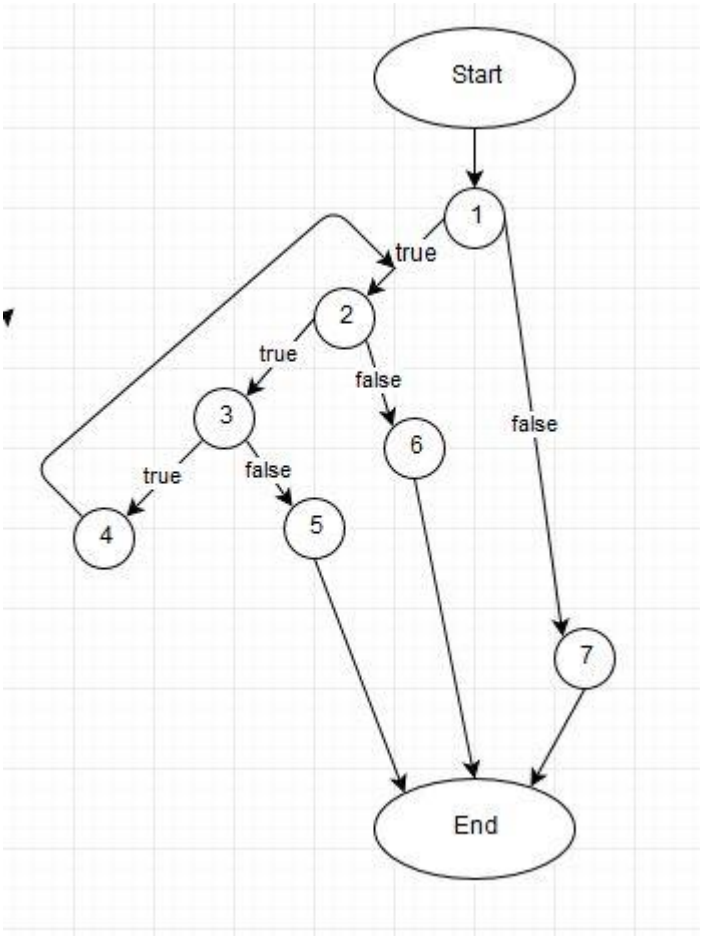
Block	Lines	Entry	Exit
1	2	2	2
2	3	3	3
3	5	5	5

```
1  begin
2  if (str.length() > 2 && str.charAt(0) == '#' && Character.isLetter(str.charAt(1))
3       return true;
4  else

5       return false

4  end
```

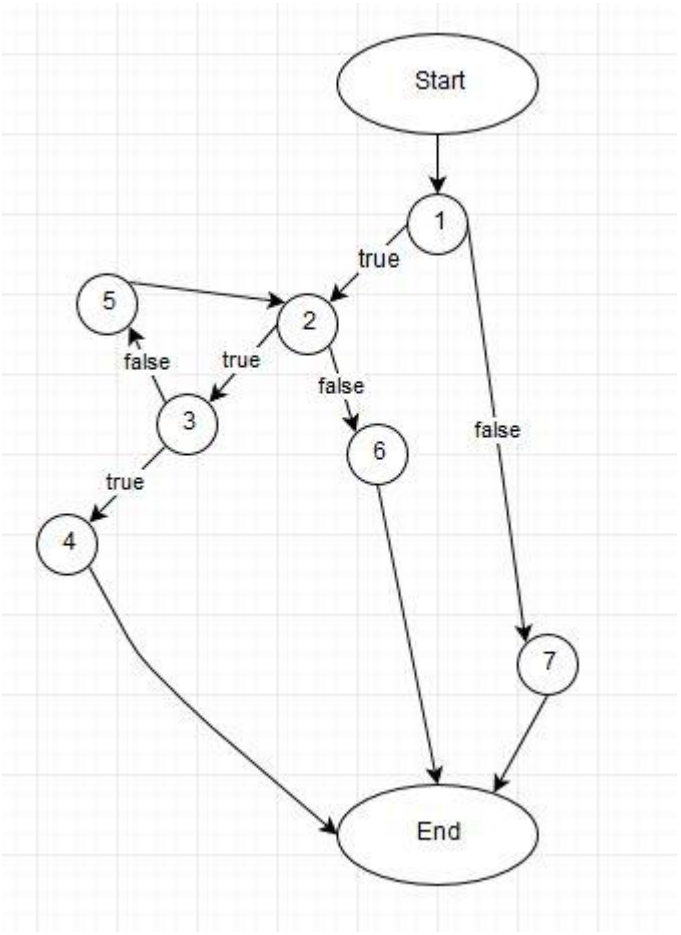
Function Name: is_num_constant



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	5	5	5
4	6	6	6
5	8	8	8
6	9	9	9
7	11	11	11

```
1 begin
2       int i = 1;
3       if (Character.isDigit(str.charAt(0)))
4               while(i <= str.length() && str.charAt(i) != '\0')
5                       if (Character.isDigit(str.charAt(i+1)))
6                               i ++;
7                       else
8                               return false;
9               return true;
10      else
11              return false;
12 end
```

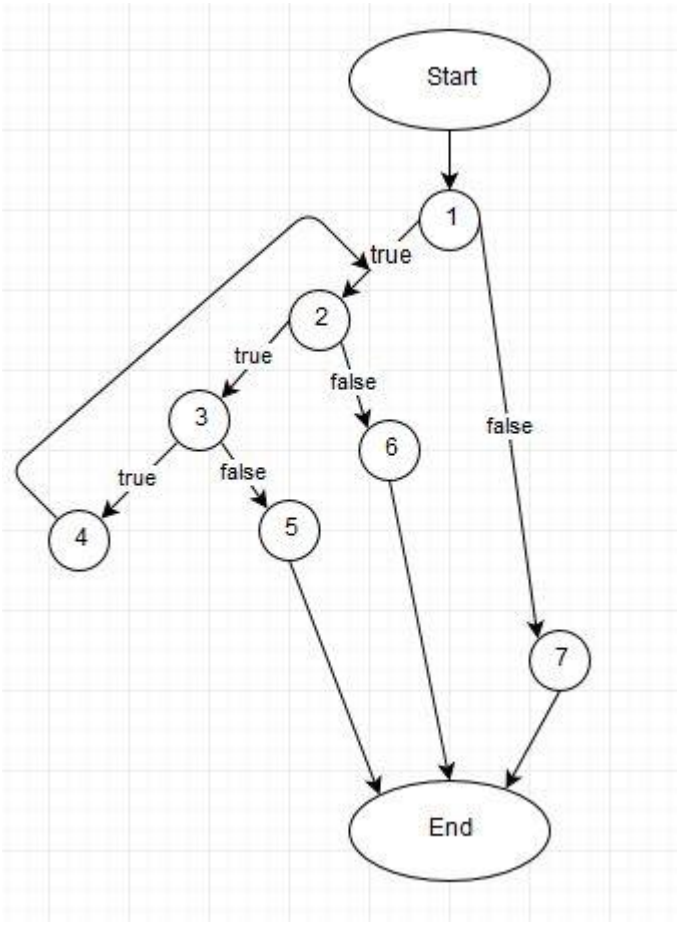
Function Name: is_str_constant



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	5	5	5
4	6	6	6
5	8	8	8
6	9	9	9
7	11	11	11

```
1  begin
2       int i = 1;
3       if (str.charAt(0) == “”)
4             while(i <= str.length() && str.charAt(i) != ‘\0’)
5                   if (str.charAt(0) == “”)
6                         return true;
7                   else
8                         i++;
9             return true;
10    else
11       return false;
12 end
```

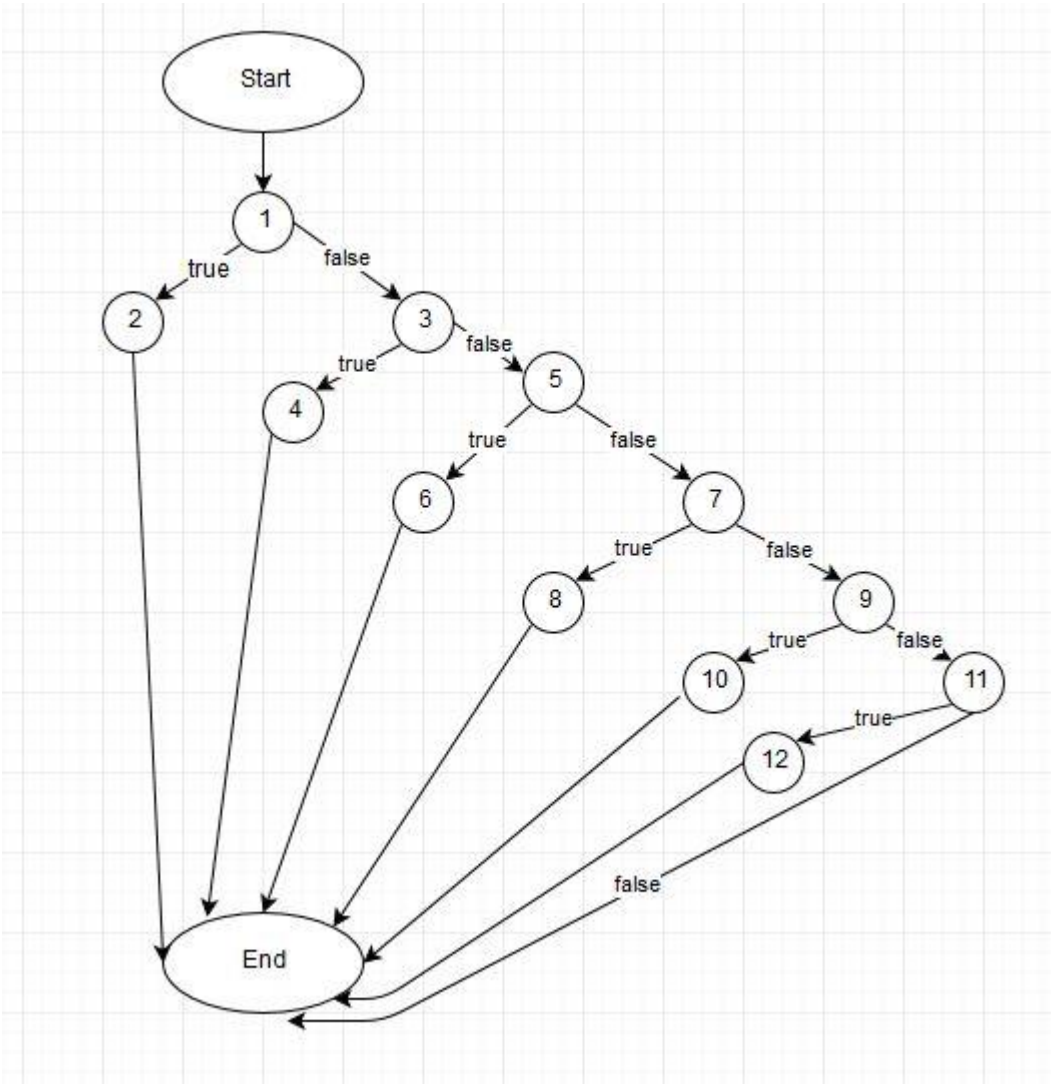
Function Name: is_identifier



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	5	5	5
4	6	6	6
5	8	8	8
6	9	9	9
7	11	11	11

```
1    begin
2         int i = 1;
3         if (Character.isLetter(str.charAt(0)))
4              while(i <= str.length() && str.charAt(i) != '\0')
5                   if (Character. isLetter (str.charAt(i+1)) || Character.isDigit(str.charAt(i)))
6                        i ++;
7                   else
8                        return false;
9              return false;
10        else
11             return true;
12 end
```

Function Name: print_spec_symbol

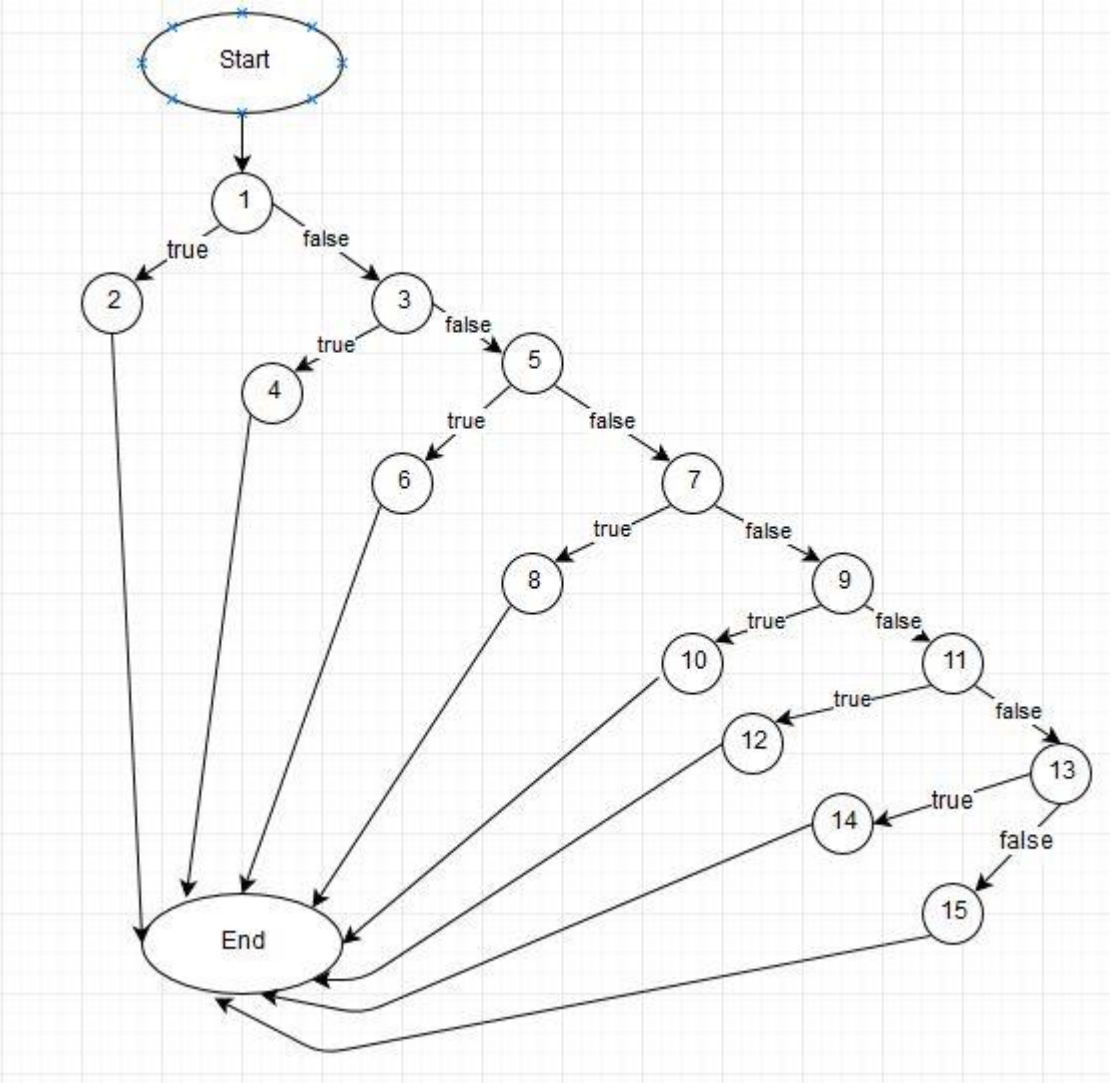


Block	Lines	Entry	Exit
1	2	2	2
2	3,4	3	4
3	5	5	5
4	6,7	6	7
5	8	8	8
6	9,10	9	10
7	11	11	11
8	12,13	12	13
9	14	14	14
10	15,16	15	16
11	17	17	17
12	18,19	18	19

```
1 begin
2         if (str.equals("{")
3            System.out.print("lparen.\n");
4            return;
5         if (str.equals(")")
6            System.out.print("rparen.\n");
7            return;
8         if (str.equals("[")
9            System.out.print("lsquare.\n");
```

```
10            return;
11        if (str.equals("]")
12            System.out.print("rsquare.\n");
13            return;
14         if (str.equals("'")
15            System.out.print("quote.\n");
16            return;
17         if (str.equals("`")
18            System.out.print("bquote.\n");
19            return;
20 end
```

Function Name: is_spec_symbol



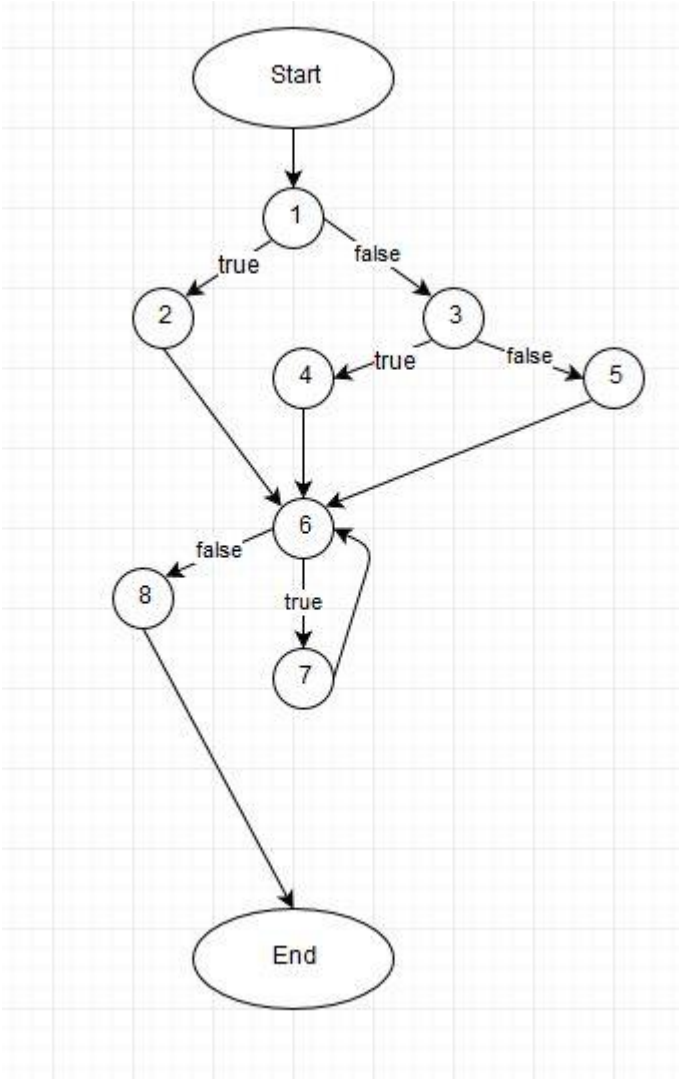
Block	Lines	Entry	Exit
1	2	2	2
2	3	3	3
3	4	4	4
4	5	5	5
5	6	6	6
6	7	7	7
7	8	8	8
8	9	9	9
9	10	10	10
10	11	11	11
11	12	12	12

12	13	13	13
13	14	14	14
14	15	15	15
15	16	16	16

```
1    begin
2        if (c == '(')
```

```
3          return true;
4       if (c == ')')
5          return true;
6       if (c == '[')
7          return true;
8       if (c == ']')
9          return true;
10      if (c == '\'')
11         return true;
12      if (c == '`')
13         return true;
14      if (c == ‘,’)
15         return true;
16      return false;
17     end
```

Function Name: main



Block	Lines	Entry	Exit
1	2,3	2	3
2	4	4	4
3	5	5	5
4	6	6	6
5	8,9	8	9

6	10,11, 12,13	10	13
7	14,15	14	15
8	16	16	16

```
1 begin
2       String fname = null;
3       if (args.length == 0)
4          fname = new String();
5       else if (args.length == 1)
6          fname = args[1];
7       else
8          System.out.print("Error!,please give the token stream\n");
9          System.exit(0);
10      Printtokens2 t = new Printtokens2();
11      BufferedReader br = t.open_token_stream(fname); /* open token stream */
12      String tok = t.get_token(br);
13      while (tok != null)
14         t.print_token(tok);
15         tok = t.get_token(br);
16      System.exit(0);
17 end
```