

print_code.1

```
2 // push state to stack and return same
3 int pushState(int id);
4 // first pop then return last state
5 int popState();
6 // true if last block was if-block
7 bool isLastOneIf(int ifs);
8
9 // print error, clear buffer and terminate program
10 void stopProgram(char *error);
11 // will remove redundant character like ' ', ',', '\t', '\n'
12 char* removeRedundant(char *source);
13 // initialize output buffer for adding more data
14 void initOutBuffer();
15
16 //print buffer and clear buffer
17 void printAndClearOutBuffer();
18 // remove leading space, ',', and add to outBuffer
19 void continueOutBuffer(char *vc);
20 // extract name and val from nameVal
21 void extractNameValue(char *nameVal, char *name, double* val);
22
23 // add variable after checking condition
24 bool addVariable(char* name, char *type);
25 // print header(1) or error(0) related to header
26 void runHeader(int check);
27 // indicate header section ends
28 void exceptHeader();
29
30 // initialize single line comment counter++
31 void initSingleComment();
32 // clear comment buffer and it's length
33 void resetBuffer();
34 // increases comment counter only
35 void process_comment();
36
37 // check if comment if closed or error if any
38 void checkForEnd();
39 // append ch to comment_buffer
40 void appendToBuffer(char ch);
41 // append string to comment_buffer
42 void appendTextToBuffer(char* ch);
43
44 // init multi-comment and setup buffer for it
45 void initMultiComment();
46 // print execution starting and changes flag
47 void initMain();
48 // print execution done and change flag
49 void stopMain();
50
51 // init var section and store last type
52 void initVarSec(char *type);
53 // called updateVariable internally
54 void assignValue(char *name, double val);
55 // call deleteVariable after removing redundant from name
56 void removeVariable(char *name);
57
58 // add variable, assign value or print error if any
59 void declareVariable(char *name, double val);
60 // extract name and val then declareVariable
61 void processVariable(char *nameVal);
62 // perform calculation on statment like a = b add 233
63 void continueVariableCalc(char *stmt, bool addVar);
64
65 // update variable value after extracing name and val
66 void updateValue(char *nameVal);
67 // delete variable with name
68 void discardVariable(char *name);
69 // true if num can be converted into number
70 bool isNumber(char* num);
71
72 // process if condition section and assign result to isLastIfValid if condition is valid
73 void processIfBody(char *body);
74 // just increase counter
75 void inc(int index);
```