1. **Description**

Let’s consider the following C++ program

#include <iostream>

using namespace std;

int main() {

int i = 1;

float b = 3.12;

float c;

while (i <= 5) {

cout << i;

++i;

}

return 0;

}

1. **Question**

A lexer has been developed to recognize the following tokens

|  |  |  |
| --- | --- | --- |
| **No** | **Token** | **Description** |
| 1 | Include | #include |
| 2 | InternalLib | <anystring> |
| 3 | Using | using |
| 4 | NameSpace | namespace |
| 5 | Semi | ; |
| 6 | Int | int |
| 7 | Integer | 2, 3, -7 |
| 8 | Main | main |
| 9 | OpenPara | ( |
| 10 | ClosePara | ) |
| 11 | Id | identiier |
| 12 | While | while |
| 13 | OpenBraket | { |
| 14 | CloseBraket | } |
| 15 | InEq | >,<,<=,>=, |
| 16 | Cout | cout |
| 17 | String | C string: “abc” |
| 18 | Inc | ++,-- |
| 19 | Return | return |
| 20 | If | if |
| 21 | Eq | ==,!= |
| 22 | Assignment | = |
| 23 | Char | ‘a’ |
| 24 | Float | -5.0, 6.7f, 9.87654e+06 |
| 25 | Shift | << |
| 26 | Flt | float |

Write a parser to parse a corresponding C++ program, assuming that the program only uses tokens involved in the example of Section I and has the following structure:

* Beginning by one/many declarations, each of which has exactly one variable with/without assigning default values. Declarations are typed integer/float.
* The next part is a while structure, whose condition is a comparison of a variable to an integer/float. This part is optional which means the program can run with/without the while structure.
* The body of the while consist of multiple instructions, whose number can be zero. The instruction can be:
  + Output a variable/integer/floating point number/string/char to the screen using cout.
  + Increase/decrease a variable using incremental/decremental operators. Both prefix and postfix notations are supported.
  + An assignment statement, which assign a value to a variable whose type can be integer/float.
* The final part is a return statement, which can return a variable/integer.