

CSE332: Compiler Design Lab

Instructions:

You are tasked to design the lexer and parser in C for a custom programming language with rules and restrictions as given below,

Rules:

1. All programs start with `#include <stdio.h>` . It's a fixed line. This line must be the beginning of the code
2. Variable names must start with an underscore, then one or more alphabets and then a one digit number followed by just one alphabet.
3. Function names must end with `Fn` and only alphabets can be used in its name.
4. `..` has to be used to end any statement.
5. Variable and function return type can be either `int` or `dec`. Variables or numbers can be returned.
6. Loops must start with a pattern like `loop_main01:` where the part between `_` and `:` has to be one or more alphabets followed by two digits. only while loop exists and it deals with only a single variable. It accepts variable type, variable name then the only comparator sign `<` and then one or more digit numbers followed by `..`
7. There is one built in function `printf` with structure like `printf(_result4m)` where the part before `,` is fixed but the part after it may contain any variable name that follows variable naming rule.
8. Custom functions like `computeValueFn(dec _val1a) { }` may or may not be present. But if they do, they will just accept one variable type followed by a variable name as input. Custom functions can be one or more and must be placed after the first line of the code and before the main function.
9. Main function is a mandatory function that must exist.

*****To design your final code, **you must use the lexer and parser code provided by me** during the lab with minor modifications only that's essential for the task. No other code will be accepted. It's not just about creating solution but also about only using specified resources for it*****

Input will be the codes that follow the rules, output will have two parts. Lexical part showing translation of elements of code to tokens and second part to show parsing of the tokenized output of first part and final verdict whether the program was accepted or not.

Valid program example for testing:

Example 1:

```
#include<stdio.h>

dec computeValueFn(dec _val1a) {
    dec _temp2x = _val1a + 5..
    return _temp2x..
}

int main() {
    dec _input3k = 10..
    dec _result4m = computeValueFn(_input3k)..

    loop_main01:
    while (dec _loopin0x < 3..) {
        printf(_result4m)..
        break..
    }

    return 0..
}
```

Example 2:

```
#include<stdio.h>

int main() {
    dec _input3k = 10..
    dec _result4m = 2..

    loop_main01:
    while (int _m7x < 3..) {
        printf(_rt4c)..
        break..
    }

    return 0..
}
```

```
}
```

Example 3:

```
#include<stdio.h>

int main() {
    dec _input3k = 10..
    dec _result4m = 2..
    printf( _rt4c)..
    return 0..
}
```