PROIECTUL 1. DEADLINE: 18 martie 2025

Să se scrie un analizor lexical pentru un limbaj de programare la alegere. Scrieți analizorul sub forma unei funcții care returnează: tipul token-ului curent, lungimea șirului corespunzător din fișierul de intrare, linia din fișierul de intrare pe care se află token-ul curent, pointer către primul caracter al token-ului curent, un mesaj de eroare atunci când este întâlnită o eroare lexicală. Funcția este apelată din programul principal, în care este citit un fișier de intrare care va fi scanat cu ajutorul acestei funcții, astfel încât să se afișeze toți token-ii care apar în fișierul de intrare (afișarea se face la consolă sau într-un fișier de ieșire). Atunci când este apelată, funcția de scanare:

-începând de la pointerul curent (care inițial indică către primul caracter al fișierului de intrare) sare peste un număr de caractere egal cu lungimea token-ului anterior (inițial această lungime este 0);

-sare peste spații, tab-uri, linii noi, până întâlnește primul caracter diferit de acestea; setează pointerul curent astfel ca să indice către acest caracter;

-identifică token-ul curent ce corespunde șirului ce începe cu caracterul depistat la pasul anterior; determină tipul acestuia și lungimea șirului corespunzător;

-în cazul în care este întalnită o eroare lexicală: semnelează aceasta printr-un mesaj; scanează fișierul de intrare în continuare până găsește primul caracter de tip spațiu, linie nouă, tab; setează pointerul curent către acest caracter; setează lungimea token-ului curent cu 0 (în felul acesta programul va afișa în continuare token-ii următori, fără să se oprească la prima eroare întâlnită).

-oprește scanarea când a întalnit sfârșitul fișierului de intrare.

- regula pe care o veți aplica în depistarea unui token este: se scanează caractere spre dreapta până se depistează cel mai lung token valid. De exemplu, pentru token-ul *ab12* urmat de un spațiu, nu ne vom opri după *a*, nici după *ab*, nici după *ab1*. Ne vom opri după *ab12*, adică atunci când întâlnim spațiu (sau un caracter diferit de literă, cifră, _)

Exemplu:

```
1.
2.
3.  int main() {
4.      int num, a12b_34, 13aaa;
5.      string x= "un string \
6.                      pe mai \" multe linii  "
7.      printf("Enter an integer: ");
8.      scanf("%d", &num);
9.
10.     // true if num is perfectly divisible by 2
11.     if(num % 2 == 0)
12.         printf("%d is even.", num);
13.     else
14.         printf("%d is odd.", num);
15.
16.     return 0;    /* comentariu ** pe /mai
17.                      multe linii   */     num++;
18.}
```

Token-ii depistați vor fi:

“int”, linia 3, key-word, 3

“main”, linia 3, key-word, 4

“(”, linia 3, separator, 1

...

“ comentariu ** pe /mai\n multe linii”, linia 16-17, comentariu,

“num”, linia 17, identificator, 3

“++”, linia 17, operator, 2

“}”, linia 18, separator, 1