**Тема 2:** Интерпретатор:

Да се напише интерпретатор на езика, определен със запазените думи:

LАBEL, GOTO, LET, READ, PRINT, IF, ENDIF, ELSE, WHILE, DONE, PUSH, POP

И оператори =, ==, !=, <, <=, >, >=, +, \*, /, -, %, &&, ||, !

Работещ над целочислени данни. Вашият интерпретатор трябва да може да работи над подадени текстови файлове, както и в REPL режим. Проверявайте за синтактични грешки и за всички такива изведете подходящи съобщения.

* С LABEL се задава етикет за безусловен преход адресиран с GOTO. Може да се разполага на произволно място в програмата.
* С LET се въвежда променлива. Всяка променлива трябва да бъде обявена преди да се използва. Областта на действие е от декларацията до края на блока.
* READ и PRINT се използват за четене на стойности от стандартния вход и извеждане на екрана на стойност.
* Конструкцията за условен преход IF има следният синтаксис:

IF <булев или целочислен израз> <блок код> [ELSE <блок код>] ENDIF

* Конструкцията за цикъл WHILE има следният синтаксис:

WHILE <условие> <блок код> DONE

* PUSH и POP съответно съхраняват един елемент в стека и изваждат стойността на един елемент от стека. Тези конструкции могат да се използват за реализиране на рекурсивни схеми, както и на масиви или други не-скаларни типове данни.

Като примери за използване на вашия интерпретатор напишете на този език програма, която въвежда числа и намира най-малкото от тях. За по-заинтересованите, напишете също и програма, която прочита определен брой числа, сортира ги и ги извежда на екрана.

**Пример за програма, която намира сума на определен брой числа:**

LET count

READ count

IF count < 0

GOTO end

ENDIF

LET sum

sum = 0

WHILE count > 0

LET x

READ x

sum = sum + x

count = count - 1

DONE

PRINT sum

LABEL end