* System Programming and Operating System (SPOS) - Assignment Number - 2

Name :- Kaustubh Shrikant Kabra.
Class :- Third Year Engineering.
Div :- A Roll Number :- 38
Batch :- T2
Department :- Computer Department
College :- AISSMS's IOIT.

★ Compare the compiler and interpreter. Give the various data structures in the design of PASS I and PASS II of a two pass Macros and of PASS I and PASS II of a two pass Assembler.

Sr. No.	Compiler	Interpreter.
1.	Compiler scans the whole program in one go.	Translates program one statement at a time.
2.	As it scans the code in one go, the error (if any) are shown at the end together.	Considering it scans code one line at a time, errors are shown line by line.
3.	Main advantage of compilers is it's execution time.	Due to interpreters being slow in executing the object code, it is preferred less.
4.	It converts the source code into object code.	It does not convert source code into object code instead it scans it line by line.

Sr. No	Compiler	Interpreter
5.	It does not require source code for later execution.	It requires source code for later execution.
6.	The compilers produce object code.	The interpreter do not produce object code.
7.	Example :- C, C++, C# etc.	Example :- Python, Ruby, Perl, SNOBOL, MATLAB, etc.

Data Structure used in Pass 1 and 2 pass assembler

① OPTAB (opcode table) -

It is used to see mnemonic operation codes and converts them to their equivalent machine language. In PASS I the OPTAB checks the operation code in the program and validate it.

② SYMTAB( ) -

It includes name and address for each label in source program. In PASS I all symbols used in the source program are added in symbol table with corresponding address and length while in PASS2 are taken from symbol table and added into the Assembly instruction to generate target program.

③ LITTAB (Literal table) -

It includes name of the literals and their corresponding addresses from the source program.

④ POOLTAB (Pool table)-

It is used to record pools in the literal tables.

PASS II of assembler-

PASS II intermediate code, SYMTAB and LITTAB and converts the intermediate code into target code with the help of SYMTAB and LITTAB entries.

1) LC = 0
2) While next statement is not END statement
   i) If START or ORIGIN statement
      a) LC = value specified in operand.
   ii) If LTORG statement-
      Update LC as per the addresses assigned to literal in current pool.
   iii) If declaration statement
      a) If DC statement then assemble the constant.
      b) LC = LC + size of operand.
   iv) If imperative statement
      a) Get operand address as per entry number from SYMTAB and LITTAB.
      b) LC = LC + size of instruction.
   v) Write target code into input line.