

• Assignment-1

• Short Questions

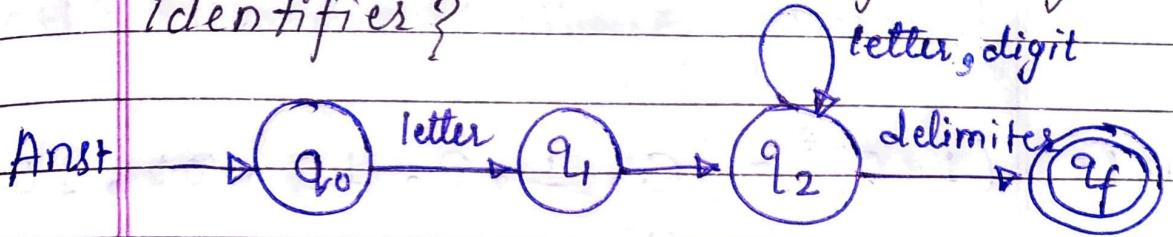
① Differentiate between dynamic loader and linker?

Ans:	Linker	Loader
(i)	The main function of linker is to generate executable files	(i) Whereas main objective of loader is to execute files to main memory
(ii)	Use of linker is to combine all object modules	(ii) It helps in allocating the address to executable code/files

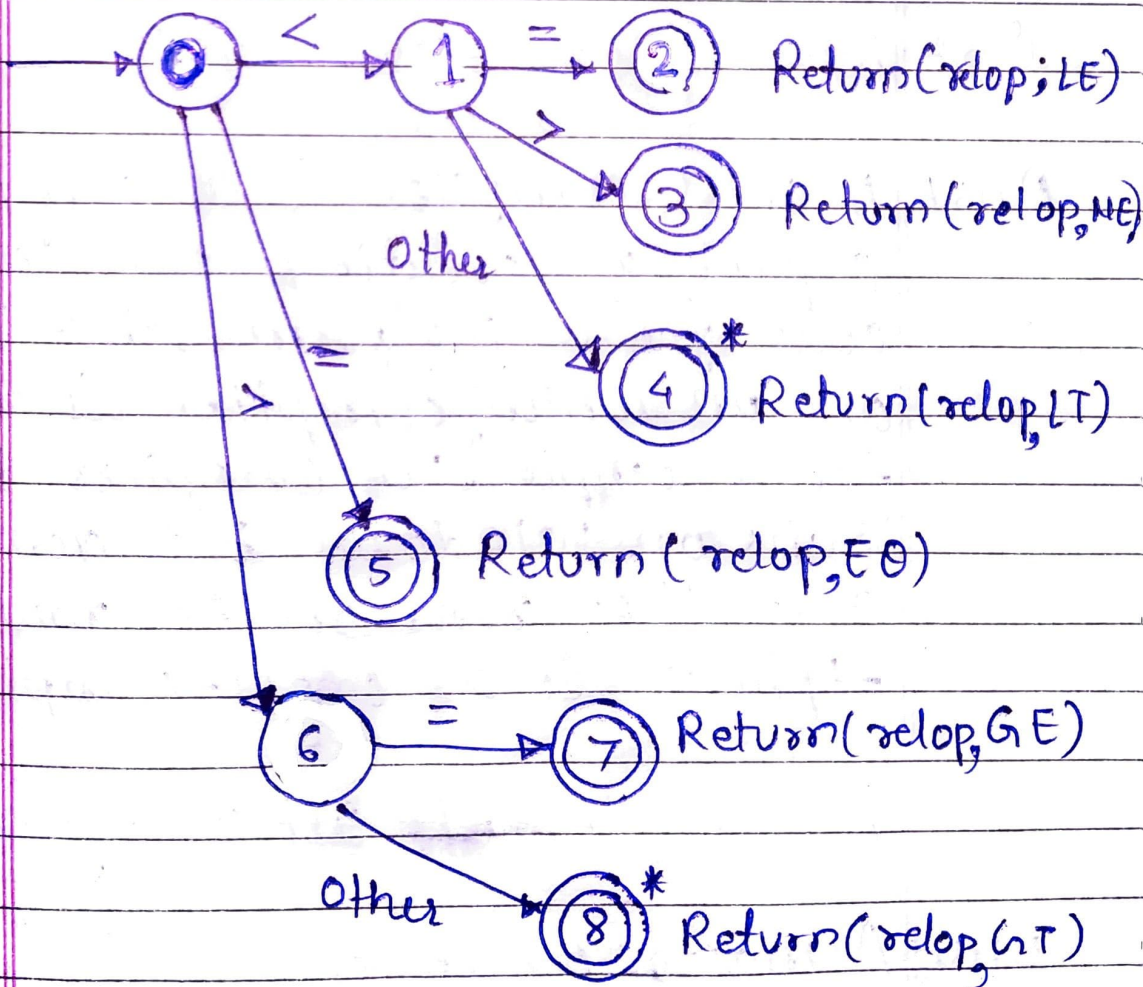
② Assembler What is Assembler?

Ans: An assembler is a program which translates assembly language program into machine code. The output of an assembler is called object file which contains a combination of machine instructions as well as the data required to place these instructions in memory.

③ Draw the transition diagram for identifiers?



Q4: Draw the transition diagram for relational operator?



Q5:- Describe the language denoted by the following regular expression $(1+0)^*$

Ans = $(1+0)^* = \{ \epsilon, 1, 0, 10, 01, 100, 11, 00, 01, \dots \}$

Set of all strings of 1's and 0's of any length including the null string.

Q6:- What is cross-compiler?

Ans:- A cross compiler is a compiler capable of creating executable code for a platform other than the one on which ^{the} compiler is running. For example, a compiler that runs on windows but generates code that runs on Android smartphone is a cross-compiler.

Q7:- Discuss the utility of macros.

Ans:- A macro (which stands for "macro-instruction") is a programming programmable pattern which translates a certain sequence of

input into a preset sequence of o/p. Macros can make tasks less repetitive by representing a complicated sequence of keystrokes.

Q8:- What do you mean by a Regular Expression?

Ans:- A regular expression is a search pattern used for matching one or more characters within a string. It can match specific characters, and ranges of characters.

Q9:- Differentiate between compilers and interpreters?

Ans:- Compiler	Interpreter
(i) Compiler scans the whole program in one go.	(i) Interpreter translates program one statement at a time.
(ii) It converts the instructions into systematic code	(ii) It doesn't convert the instructions. Instead it directly works on source language

Q104 Discuss the merits and demerits of single pass compilers and multi-pass compilers?

Ans: A single pass compiler is a compiler that transfers through the reference code of each compilation unit for only once. A multi-pass compiler is a type of compiler that prepares the reference code or general syntax tree of performance numerous times.

Single pass Compiler

→ Advantages + More effective than multipass compilers in the compilers point of view

→ Disadvantages + It compiles less efficient programs.

Multi-pass Compiler

→ Advantages + It can be played very role useful & when optimizing code.

→ Disadvantages - It is a very slower process which takes a lot of time to compile the code.

Q11 - Describe various compiler writing tools?

Ans - The various compiler writing tools ~~are~~ which ~~are~~ aid in the construction of compilers -

(i) Parser Generator: e.g. Yacc

(ii) Code generator generators

(iii) Scanner generators

- e.g. lex

- The I/p of lex consists of a definition of each token as a regular expression

(i) Parser Generator -

It is an application which generates a parser. Sometimes also called a 'compiler compiler'.

(ii) Code Generator Generator -

It is a process by which a compiler's code generator converts some intermediate representation of source code

into a form (for eg. machine code) that can be readily executed by machine

(iii) Scanner Generator

It generates lexical analyzers from the input that ~~contains~~ consists of regular expression description based on tokens of a language.

Q121 Explain the term bootstrapping with example?

Ans: Bootstrapping is a process in which simple language is used to translate more complicated program which in turn may handle for more complicated program.

This complicated program can further handle even more complicated program and so on.

Example:-

We can create compilers of many different forms. Now we will generate

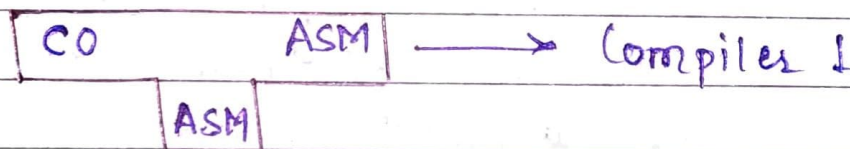
X = New language
 Y = Implemented language
 Z = generated language



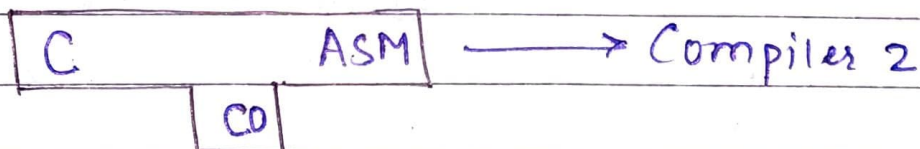
These two languages must be same

These two languages must be same

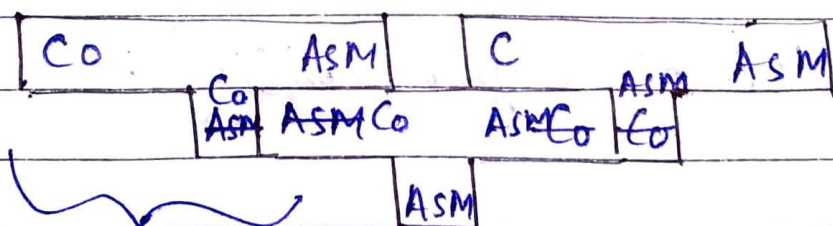
Step 1: first we write a compiler for a small ϕ of C in assembly language



Step 2: Then using with small subset of C i.e. $C\phi$, for the source language C , the compiler is written



Step 3: Finally we get compile the second compiler. Using compiler 1 & 2 is compiled



Compiler 2

Compiler 1

Q131 What is the role of lexical analyser? Enumerate the issues handled by lexical analyzer?

Ans As the first phase of a compiler, the main task of the lexical analyzer is to read the input characters of the source program, group them into lexemes, and produces as output a sequence of tokens for each lexeme in the source program.

The issues handled by lexical analyzer is that:

- (i) Simpler design is the most important consideration. The separation of lexical analysis from syntax analysis often allows us to simplify one or more these phase
- (ii) Compiler efficiency is improved
- (iii) Compiler portability is enhanced



PAGE NO.: _____

DATE: / /

Q14: Differentiate between linker and loader?

Ans: Linker

loader

(i) The main function of linker is to generate exe files

(ii) Use of linker is to combine all object module

(i) The main objective of loader is to execute files to the main memory

(ii) It helps in allocating the address to executable code i files