QA
10/04/2024.

Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

Assignment No: 1

Q.1 With references to assembler, consider any assembly program and generate Pass-1 and Pass-2 output. Also show contents of Database table involved in it.

⇒

- Assembly program:

```
START   100
        LDA  ALPHA
        STA  BETA
        HLT
ALPHA   RES  1
BETA    RES  2
        END  START
```

- Pass-1 output (Symbol table):

Symbol	Address
START	100
ALPHA	101
BETA	102

- Pass-2 output (Machine code):

100	0000	1010
101	0001	1010
102	0010	1010

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- **Database table:**

Address	Label	Opcode	Operand
200	START		
201		LDA	ALPHA
202		STA	BETA
203		HLT	
204	ALPHA	RES	1
205	BETA	RES	1

- In Pass-1, the assembler reads the assembly code, assigns addresses to symbols, and creates a symbol table.
- In Pass-2, the assembler translates assembly instruction into machine code using the symbol table generated in Pass-1.
- The database table shows the address, label, opcode and operand for each instruction or data declaration in the program.

Q.2. Write short note on : YACC

⇒ Each translation rule input to YACC has a string specification that resembles a production of grammar it has a non-terminal on the LHS and a few alternatives on the RHS.

- For simplicity, we will refer to a string specification as a production. YACC generates an LALR(1) parser for language L from the productions, which is a bottom-up parser.
- The parser would operate as follows : For a shif action, it would invoke the scanner to obtain the next token and continue the parse by using token.
- While performing a reduced action in accordance with production, it would perform the sementic action associated with that production.
- The semantic actions associated with productions achieve the building of an intemediate representation or target code as follows :

1) Every non-terminal symbol in the parser has an attribute.
2) The semantic action uses the values of these attributes for building the intermediate representation or target code.

- A parser generator is a program that takes as input a specification of a syntax and produces as a output procedure for recognizing that language. Historically, they are also called compile compilers

- Input File: YACC input file is divided into three parts.

/* definition */
%%
/* rules */
%%
/* auxiliary routines */

- Definition Part: The definition part includes the information about the tokens used in syntax definition. The definition part can include C code external to the definition of the parser and variable declarations, within %{ and %} in the first column.

- Rule part: The rule part contains grammar definitions in a modified BNF form. Actions are C code in { } and can be embedded inside.

- Auxiliary Routine Part: The auxiliary routine part is only C code. It includes function definition for every function needed in the rules part.