1. Write a program to generate symbol table of two pass assembler.

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 | SAMPLE | START | 0 |
| 2 |  | USING | \*, 15 |
| 3 |  | A | 1, FOUR |
| 4 | TEMP | EQU | 10 |
| 5 | FOUR | DC | F ‘4’ |
| 6 |  | END |  |

1. Write a program to generate literal table of two pass assembler.

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 | SAMPLE | START | 0 |
| 2 |  | USING | \*, 15 |
| 3 |  | A | 1, FOUR |
| 4 |  | A | 1, =F’3’ |
| 5 | FOUR | DC | F ‘4’ |
| 6 |  | A | 2, =F’5’ |
| 7 |  | END |  |

1. Write a program to generate base table of two pass assembler

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 | SAMPLE | START | 0 |
| 2 |  | USING | \*, 15 |
| 3 |  | A | 1, FOUR |
| 4 | TEMP | EQU | 10 |
| 5 | FOUR | DC | F ‘4’ |
| 6 |  | END |  |

1. Write a program to generate MDT for macro processor

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 |  | MACRO |  |
| 2 |  | ADDM | &arg0, &arg1 |
| 3 |  | A | 1, &arg0 |
| 4 |  | A | 2, &agr1 |
| 5 |  | MEND |  |
| 6 | DATA1 | DC | F’4’ |
| 7 | DATA2 | DC | F’5’ |
| 8 |  | ADDM | DATA1, DATA2 |
| 9 |  | END |  |

1. Write a program to generate MNT for macro processor

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 |  | MACRO |  |
| 2 |  | ADDM | &arg0, &arg1 |
| 3 |  | A | 1, &arg0 |
| 4 |  | A | 2, &agr1 |
| 5 |  | MEND |  |
| 6 | DATA1 | DC | F’4’ |
| 7 | DATA2 | DC | F’5’ |
| 8 |  | ADDM | DATA1, DATA2 |
| 9 |  | END |  |

1. Write a program to generate ALA for macro processor

|  |  |  |  |
| --- | --- | --- | --- |
| stmt no | Symbol | opcode | operands |
| 1 |  | MACRO |  |
| 2 |  | ADDM | &arg0, &arg1 |
| 3 |  | A | 1, &arg0 |
| 4 |  | A | 2, &agr1 |
| 5 |  | MEND |  |
| 6 | DATA1 | DC | F’4’ |
| 7 | DATA2 | DC | F’5’ |
| 8 |  | ADDM | DATA1, DATA2 |
| 9 |  | END |  |

1. Write a program to compute first of given grammar.
2. Write a program to compute follow of given grammar.
3. Write a Lex Program to number of letters, words, digit, lines and black spaces in given file.
4. Write a Lex program to recognize valid identifiers, keywords in given input file.
5. Write a Lex program to recognize a word starting with ‘a’ and ending with ‘b’.
6. Write a LEX YACC program to recognize a valid arithmetic expression.
7. Write a LEX YACC Program to implement scientific calculator.
8. Write a program to implement common sub expression elimination optimization technique.
9. Write a program to implement dead code elimination optimization technique.
10. Write a program to implement 3 address code.
11. Write a program to implement code generation.