**(C) 1.** Write a program to generate Symbol table of a two-pass Assembler for the given Assembly language source code.

|  |
| --- |
| **INPUT/CODE**  START 180  READ M  READ N  LOOP MOVER AREG, M  MOVER BREG, N  COMP BREG, =’200’  BC GT, LOOP  BACK SUB AREG, M  COMP AREG, =’500’  BC LT, BACK  STOP  M DS 1  N DS 1  END |

**++++++++++++++++++++++++++++++++++++**

**(4)**

**(A)2.** Write a program to generate Literal table of a two-pass Assembler for the given Assembly language source code.

|  |
| --- |
| **INPUT/CODE**  START 100  READ A  READ B  MOVER AREG, =’50'  MOVER BREG, =’60’  ADD AREG, BREG  LOOP MOVER CREG, A  ADD CREG, ='10'  COMP CREG, B  BC LT, LOOP  NEXT SUB AREG, ='10'  COMP AREG, B  BC GT, NEXT  STOP  A DS 1  B DS 1  END |

**(A)3**.Write a program to generate Pool table of a two-pass Assembler for the given Assembly language source code.

|  |
| --- |
| **INPUT/CODE**  START 100  READ A  MOVER AREG, ='1'  MOVEM AREG, B  MOVER BREG, ='6'  ADD AREG, BREG  COMP AREG, A  BC GT, LAST  LTORG  NEXT SUB AREG, ='1'  MOVER CREG, B  ADD CREG, ='8'  MOVEM CREG, B  PRINT B  LAST STOP  A DS 1  B DS 1  END |

**++++++++++++++++++++++++++++++++++++**

**(A)4.**Write a program to generate Intermediate code of a two-pass Assembler for the given Assembly language source code.

|  |
| --- |
| **INPUT/CODE**  START 100  READ A  READ B  MOVER AREG, A  SUB AREG, B  STOP  A DS 1  B DS 1  END |

**++++++++++++++++++++++++++++++++++++**

**(11)**

**(B++++++++++++++++++++++++++++++++++++**

**(13)**

**(A)5.**Write a program to generate Intermediate code of a two-pass Macro processor.

|  |  |
| --- | --- |
| **INPUT/CODE**  LOAD A  MACRO ABC  LOAD p  SUB q  MEND  STORE B  MULT D  MACRO ADD1 ARG  LOAD X  STORE ARG  MEND  …continued… | …continued…  LOAD B  MACRO ADD5 A1, A2, A3  STORE A2  ADD1 5  ADD1 10  LOAD A1  LOAD A3  MEND  ADD1 t  ABC  ADD5 D1, D2, D3  END |

**++++++++++++++++++++++++++++++++++++**

**(14)**

**++++++++++++++++++++++++++++++++++++**

**++++++++++++++++++++++++++++++++++++**

**(16)**

**(D)5.**Write a program to generateIntermediate code of a two-pass Macro processor.

|  |  |
| --- | --- |
| **INPUT/CODE**  LOAD J  STORE M  MACRO EST  LOAD e  ADD d  MEND  LOAD S  MACRO SUB4 ABC  LOAD U  STORE ABC  MEND | …continued….  LOAD P  ADD V  MACRO ADD7 P4, P5, P6  LOAD P5  SUB4 XYZ  SUB 8  SUB 2  STORE P4  STORE P6  MEND  EST  ADD7 C4, C5, C6  SUB4 z  END |

**++++++++++++++++++++++++++++++++++++**

**(17)**

**(A)6.**Write a program to generate MDT MNT(Macro Definition Table) of a two-pass Macro processor.

|  |  |
| --- | --- |
| **INPUT/CODE**  LOAD A  STORE B  MACRO ABC  LOAD p  SUB q  MEND  MACRO ADD1 ARG  LOAD X  STORE ARG  MEND  ….Continued…. | …continued….  MACRO ADD5 A1, A2, A3  STORE A2  ADD1 5  ADD1 10  LOAD A1  LOAD A3  MEND  ABC  ADD5 D1, D2, D3  END |

**++++++++++++++++++++++++++++++++++++**

**(18)**

**(19)**

**(++++++++++++++++++++++++++++++++++++**

**(20)**

**(B)7.**Write a program to generate MDT MNT(Macro Name Table) of a two-pass Macro processor.

|  |  |
| --- | --- |
| **INPUT/CODE**  LOAD J  STORE M  MACRO EST1  LOAD e  ADD d  MEND  MACRO EST ABC  EST1  STORE ABC  MEND | MACRO ADD7 P4, P5, P6  LOAD P5  EST 8  SUB4 2  STORE P4  STORE P6  MEND  EST  ADD7 C4, C5, C6  END |

**++++++++++++++++++++++++++++++++++++**

**(21)**

**(A)8.**Write a program using LEX Tool, to implement a lexical analyzer for parts of speech for given English language without Symbol table.

**INPUT**

Dread it. Run from it.

Destiny arrives all the same.

**(A)9.**Write a program using LEX Tool, to implement a lexical analyzer for given C programming language without Symbol table.

**INPUT**

{

int m=10,n=2,o;

o = m – n;

}

**(D)9.**Write a program using LEX Tool, to implement a lexical analyzer for given C programming language without Symbol table.

**INPUT**

{

int a=3;

int b=4;

float c;

c = (a\*a + b\*b) \*2

}

**++++++++++++++++++++++++++++++++++++**

**(31)**

**(E)9.** Write a program using LEX Tool, to implement a lexical analyzer for given C programming language without Symbol table.

**INPUT**

{

int total =100;

inti=10;

printf("The value of total and i is : %d, %d", total, i);

}

**++++++++++++++++++++++++++++++++++++**

**(32)**

**(A)10.**Write a program to evaluate a given arithmetic expression using YACC specification.

**INPUT**

0.33\*12-4-4+(3\*2)

**++++++++++++++++++++++++++++++++++++**

**++++++++++++++++++++++++++++++++++++**

**(35)**

**++++++++++++++++++++++++++++++++++++**

**(36)**

**(A)11.**Write a program to evaluate a given variable name using YACC specification.

**SAMPLE INPUT**

1. pune
2. PUNE
3. Pune1
4. pUNE\_2

**++++++++++++++++++++++++++++++++++++**

**(38)**

**(C)11.**Write a program to evaluate a given built-in functions using YACC specification.

**INPUT**

1.u= sqrt(36)

2. v = strlen(“pune”)

**++++++++++++++++++++++++++++++++++++**

**(39)**

**(D)11.**Write a program to evaluate a given built-in functions using YACC specification.

**INPUT**

u= sin(12)+cos(12)

**++++++++++++++++++++++++++++++++++++**

**(40)**

**(E)11.**Write a program to evaluate a given built-in functions using YACC specification.

**INPUT**

p= pow(3,2) / log (24)

**++++++++++++++++++++++++++++++++++++**

**(41)**

**(A)12.**Write a program to generate three address code for the given simple expression.

**INPUT**

w = u\*u - u\*v+ v\*v