

1. What you mean by EXTRN and ENTRY Assembler Pseudo ops?
2. Draw a Flowchart for two pass Assembler.
3. Develop an algorithm for one pass macro processor with no nested calls?
4. For the following Assembly Program:

```

PRGAM2    START    0
          USING    *,15
          LA      15,SETUP
          SR      TOTAL,TOTAL
AC        EQU      2
INDEX     EQU      3
TOTAL     EQU      4
DATABASE  EQU      13
SETUP     EQU      *
          USING    SETUP,15
          L       DATABASE, = A(DATA1)
          USING   DATAAREA,DATABASE
          SR      INDEX, INDEX
LOOP      L       AC, DATA1 ( INDEX)
          AR      TOTAL, AC
          A       AC, = F'5'
          ST      AC, SAVE (INDEX)
          A       INDEX, = F'4'
          C       INDEX, = F'8000'
          BNE     LOOP
          LR      1,TOTAL
          BR      14
          LTORG
SAVE      DS      2000F
DATAAREA  EQU      *
DATA1     DC      F'25,26,97,...2000 Nos
          END

```

Write

- a) Symbol Table
- b) Literal Table
- c) Base table

d) Machine Hex code (Take the hex opcode of every instruction as 'AB')

5. Expand the following macro .

```
MACRO
SS    &ss1, &ss2,&ss3
ST    &ss1,SAVE
L     &ss2,POINTER
AR    &ss3,&ss3
SR    &ss2,&ss2
MEND
START
ENTRY RESULT
EXTRN SUM
BALR  12,0
USING *,12
SS    14,2,5
SAVE  DS    F
RESULT DS    F
POINT DC    H'5'
END
```

Make the following

- (a) MDT
- (b) MNT
- (c) ALA
- (b) Expand the Macro

For the above expanded program, convert the program into machine binary code, Assume the hex op-code of all the instructions as '11'.

Make the following

- (a) POT
- (b) MOT
- (c) ST & LT
- (d) Machine Hex Code