Macro

```
abc macro x
  if x eq BL
    mov dl,65
  else
    mov dl,66
  endif
endm
.model small
.code
 Mov  AH,2
 abc  BL
 int   33
 abc  CL
 int   33
 Mov AH,76
 int   33
END
```

Register name as parameter o/p AB

```
aa macro x
  x dl,bl
endm
.model small
.code
  Mov ah,2
  Mov bl,30
  aa mov
  aa add
  aa add
  Int    33
  Mov AH,76
  Int    33
END
```

Instruction name as parameter o/p Z

```
abc macro x
  Mov dl,65+x
  Int  33
  Int  33
endm
pqr macro x
  Mov dl,66
  Int  33
endm
ttt macro x
  x 5
endm
.model small
.code
 Mov AH,2
 ttt   abc
 ttt    pqr
 Mov AH,76
 Int    33
END
```

Macro name can be parameter o/pFFB

```
.model small
.code
Mov  AH,2
Mov  DL.0
n=65
Jmp   L1
n=n+1
Add  DL,5
L1:
Add   DL,n
Int     33
Mov  AH,76
Int     33
End
```

o/p B

Macro expansion and execution are different phases o/pB (notA)

```
abc macro
local n
  n=70
  Mov  DL,n+3
  Int  21h
  n=80
endm
.model small
.code
  n=65
  Mov  AH,2
  abc
  Mov  DL,n
  Int     33
  Mov  AH,76
  Int     33
END
```

o/pIA

IP(n not local)

4. ABCD;ABCD;ABCD;

5. F,EE,DDD,CCCC;

6. ABC,AB,A
BCD,BC,B

```
.model small
.Code
 Mov AH,2
 Mov DL,66
 Rept 5
   Int 33
 endm
 Mov AH,76
 Int 33
End
```

o/p BBBBB

1. BBBBBaaaaa

2. BCDEF (add)

3. BaBaBaBa

```
Mov AH,2
t=5
rept 3
  Mov DL,t+65
  rept t
    Int 33
  endm
  t=t-1
  Mov DL,';'
  Int 33
endm
Mov AH,76
Int 33
```

o/p FFFFF;EEEE;DDD;

15 Letter + digit b5→g s2→u (Inc)

13 Read a digit & print d A's 4→AAAA

16 ascii mod 3 assume ascii between 70 and 90

14 Read a digit print double (assume <~~$~~5) 4→8

13,14,15 Rept Mov Int + = CMP JE

```
abc macro n
    Mov   DL,n
endm
.model small
.code
  Mov  AH,2
  Mov  CL,90
  abc   CL
  Int    33
  Mov  DL,CL
  Mov  DL,90
  Mov  DL,cs:[9]
  Int    33
  Mov  DL,cs:[11]
  Int    33
  Mov  DL,cs:[5]
  Int    33
  Mov  AH,76
  Int    33
End
```

o/pZ(209)Z(209)

abc CL is Mov DL,CL not Mov DL,90

```
.model small
.code
  m=50
  t=65
  if m gt t
    Mov  CL,kk
    t=t+1
  else
    Mov  DL,t
  endif
  Mov  AH,2
  Int    33
  Mov  AH,76
  Int    33
End
```

The above program does not show error because of Mov CL,kk but the next program shows.

opA

```
.model small
.code
  Mov BL,50
  t=65
  Cmp BL,t
  JNG  L
  Mov  CL,kk
  t=t+1
  Jmp  M
L:Mov  DL,t
M:Mov AH,2
  Int   33
  Mov AH,76
  Int   33
END
```

Replace ~~kk~~ by 12 φB because macro expansion is done first

```
ttt macro k
  Mov DL,k
  Int  33
  Int  33
endm
ppp macro t
  local ttt
  ttt macro k
    Mov dl,k+t
    Int  33
  endm
  ttt 66
endm
.model small
.code
  Mov  Ah,2
  ttt 65
  ppp 6
  ttt 65
  Mov  AH,76
  Int 33
end
```

o/pAAHAA

AAHG (No Local)

```
ttt macro
  Mov  DL,65
  Int  33
endm
ppp macro
  Mov  DL,66
  Int  33
endm
abc macro
  local ttt
  ttt macro
    ppp macro
      Mov  DL,67
      Int  33
    endm
    Mov  DL,97
    Int  33
  endm
  ttt
  Mov DL,90
  Int 33
endm
```

```
.model small
.code
  Mov AH,2
  ttt
  ppp
  abc
  ppp
  ttt ← ppp
  Mov AH,76
  Int   33
END
```

ABaZCAC

ttt →ppp
ABBZBAB

no local
ABaZCaC
ttt →ppp
ABBZBaC

mov reg$_1$,reg$_2$ 10001010 11reg$_1$reg$_2$ CL:001   DL:010   Mov reg,number 10110reg number

Replace 9,11,5 by 8,10,4 respectively output is Z(138)(178)(138)

On following NO Jump of any type use Rept Mov Int Sub add adc

7. Read ~~15~~5 letters How many are less than 100 Dahkb→2 8. No Adc (Use CBW)

9. Read a letter ascii mod 7 [CBW NOT AND] A→2 10. ascii div 7 A→9 ~~CBW~~

1~~0~~1. Read 2 digits output product 98→H 95→- [CBW And]

12. Read a digit X o/p 1+2+3+ .. +X+50 2→5 6→G