Tables of the macro preprocessor

Table	Fields in each entry
Macro name table (MNT)	- Macro name, - No. of positional parameters (#PP) - No. of keyword parameters (#KP) - No. of expansion time variables (#EV) - MDT pointer (MDTP) - KPDTAB pointer (KPDTP) - SSTAB pointer (SSTP)
Parameter Name Table (PNTAB)	parameter name
EV Name Table (EVNTAB)	EV name
SS Name Table (SSNTAB)	SS name
Keyword Parameter Default Table (KPDTAB)	parameter name, default value
Macro Definition Table (MDT)	Label, Opcode, Operands
Actual Parameter Table (APTAB)	Value
EV Table (EVTAB)	Value
SSTable (SSTAB)	MDT entry #

```
      MACRO
      CLEARMEM  &X , &N , &REG=AREG
       LCL      &M
&M     SET      0
       MOVER    &REG, ='0'
.MORE  MOVEM    &REG, &X+&M
&M     SET      &M+1
       AIF      (&M NE &N) .MORE
       MEND
```

PNTAB

X
N
REG

EVNTAB

M

SSNTAB

MORE

MNT

name	#PP	#KP	#EV	MDTP	KPDTP	SSTP
CLEARMEM	2	1	1	25	10	5

KPDTAB

10	REG	AREG

SSTAB

5	28

MDT

```
25          LCL     (E,1)
26  (E,1)    SET      0
27          MOVER   (P,3), ='0'
28          MOVEM   (P,3), (P,1)+(E,1)
29  (E,1)    SET     (E,1)+1
30          AIF     (E,1) NE (P,2)) (S,1)
31          MEND
```

CALL ⇒ CLEARMEM AREA, 10

APTAB

AREA
10
AREG

EVTAB

0

Advanced Macro Facility

```
MACRO
EVAL  &P, &Q, &R
AIF (&Q EQ &P) .ONLY
MOVER  AREG, &P
 SUB  AREG, &Q
 ADD  AREG, &R
 AGO  .OVER
.ONLY  MOVER  AREG, &R
.OVER  MEND
```

1st Macro defn

```
MACRO
CLEAR  &X, &N, &REG = AREG
  LCL  &M
&M  SET  0
   MOVER  &REG, ='0'
.MORE  MOVEM  &REG, &X+&M
 &M  SET  &M+1
   AIF (&M NE &N) .MORE
 MEND
```

2nd Macro defn

```
CLEAR  B, 3
EVAL  10, 5, 2
EVAL  10, 10, 7
```

calls

Macro Expansion

1. Perform initializations for the expansion of a macro
   a) MEC = MDTP field of MNT entry
   b) Create EVTAB with # EV entries & set EVTAB_ptr.
   c) Create APTAB with #PP + #KP entries & set APTAB_ptr.
   d) Copy keyword parameter defaults from the entries KPDTAB[KPDTP] ---- KPDTAB[KPDTP + #KP - 1] into APTAB[#PP + 1] ----- APTAB[#PP + #KP]
   e) Process positional parameters in the actual parameter list & copy them into APTAB[1] --- APTAB[#PP]
   f) Process keyword parameters
2. While statement pointed by MEC is not MEND statement.
   a) If a model statement then
      i) Replace operands of the form (P, #n) & (E, #m) by values in APTAB[n] and EVTAB[n] resp.
      ii) Output the generated statement
      iii) MEC = MEC + 1;
   b) If a SET statement with the spec: (E, #m) in label field then
      i) Evaluate the expression in the operand field & set an appropriate value in EVTAB[m]
      ii) MEC = MEC + 1
   c) If an AGO statement with (S, #s) in operand field then
      MEC = SSTAB[SSTP + s - 1]
   d) If an AIF with (S, #s) in operand field
      If cond^n in AIF is true then
      MEC = SSTAB[SSTP + S - 1]
3. Exit

```
START 100
MACRO
 M1 &A , &B , &C=
 -----
MEND
MACRO
M2 &D , &F=10
 ----
MEND
MACRO
M3 &A, &C, &F=5
 -----
MEND
MACRO
M4 &X , &Y= , &Z=
 ----
MEND
MACRO
M5 &P , &Q , &R
 ----
MEND

 M2  100
 M1  5 , 10 , &C=9
 M2 50 , &F=11
 M3  21 , 22
 M4  5 , &Y=10 , &Z=11
 M3  21 , 22 , &F=6
 M5  7 , 8 , 9

END
```

M1 M2 M3 M4 M5

Calls

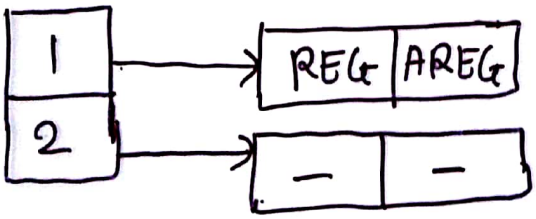
```
MACRO
INCR_D  &MEM_VAL, &INCR_VAL, &REG=AREG
MOVER   &REG, &MEM_VAL
ADD     &REG, &INCR_VAL
MOVEM   &REG, &MEM_VAL
MEND

MACRO
COMPUTE &FIRST, &SECOND
  MOVEM  BREG, TMP
INCR_D  &FIRST, &SECOND, REG=BREG   <- call to INCR_D (nested call)
  MOVER  BREG, TMP
  MEND

  COMPUTE  AREA, 3   <- call to COMPUTE
END.
```

Data structures

MNT

	name	#PP	#KP	MDTP	KPDTP
1	INCR_D	2	1	1	1
2	COMPUTE	2	0	5	2

KPDTAB



APTAB for INCR_D call (Inner call)

1	FIRST	(P,1)
2	SECOND	(P,2)
3	BREG	(P,3)

APTAB for COMPUTE (outer)

1	AREA
2	3

```
MACRO
INCR_D  &MEM_VAL, &INCR_VAL=, &REG=AREG
 MOVER  &REG, &MEM_VAL
 ADD    &REG, &INCR_VAL
 MOVEM  &REG, &MEM_VAL
MEND
```

```
MACRO
COMPUTE
MACRO   &FIRST, &SECOND, &THIRD = 100
 MOVEM  BREG, &THIRD
 INCR_D &FIRST, &INCR_VAL=101, &REG=BREG
 INCR_D 500, &INCR_VAL=501
 MOVER  BREG, &THIRD
 ADD    BREG, &SECOND
MEND
```

nested calls

```
INCR_D  AREA, &INCR_VAL=3
COMPUTE AREA, 3
END
```

} calls.

---

1. Show the contents of PNTAB, MNT, MDT, KPDTAB by processing macro definitions.

2. Show the contents of APTABs by processing macro calls. Also write the expanded code.

```
MACRO
DEFINE  &XYZ
 MACRO
 &XYZ   &X  &Y  &OP
  MOVER  AREG, &X
  &OP    AREG, &Y
  MOVEM  AREG, &X
 MEND
MEND
```

Inner Macro defn.

Outer macro defn

```
MACRO
COMPUTE &F, &S
MOVEM   BREG, TMP
INCRM   &F, &S, BREG
MOVER   BREG, TMP
MEND
```

← nested call

```
MACRO
INCRM  &M, &I, &R
 MOVER  &R, &M
 ADD    &R, &I
 MOVEM  &R, &M
MEND
START  100
 DEFINE   CACL
COMPUTE   X, Y
 CALC     A, B, MULT
END
```

← call to DEFINE
← call to COMPUTE
← call to CALC

Early expansion

MNT

	name	#PP	#KP	MDTP	KPDTP
1	INCR_D	1	2	1	1
2	COMPUTE	2	1	5	3

KPDTP

	name	Default value
1	INCR_VAL	—
2	REG	AREG
3	THIRD	100

MDT

INCR_D →	1	MOVER (P,3), (P,1)
	2	ADD (P,3), (P,2)
	3	MOVEM (P,3), (P,1)
	4	MEND
COMPUTE →	5	MOVEM BREG, (P,3)
1st call	6	MOVER BREG, (P,1)
	7	ADD BREG, 101
	8	MOVEM BREG, (P,1)
2nd call	9	MOVER AREG, 500
	10	ADD AREG, 501
	11	MOVEM AREG, 500
	12	MOVER BREG, (P,3)
→	13	ADD BREG, (P,2)
	14	MEND

Expanded code

1st call

INCR_D AREA, &INCR_VAL=3

APTAB

1 - AREA
2 - 3
3 - AREG

+MOVER AREG, AREA
+ADD AREG, 3
+MOVEM AREG, AREA

2nd call ⇒ COMPUTE AREA, 3

+ MOVEM BREG, 100
+ MOVER BREG, AREA
+ ADD BREG, 101
+ MOVEM BREG, AREA
+ MOVER AREG, 500
+ ADD AREG, 501
+ MOVEM AREG, 500
+ MOVER BREG, 100 + ADD BREG, 3

APTAB

1 - AREA
2 - 3
3 - 100

Keyword parameter processing in Pass II of macro preprocessor

```
MACRO
 M1  &x, &y, &A=AREG, &B=BREG
 MOVER &A, &x
  ADD &A  ='1'
 MOVER  &B = &y
  ADD  &B, ='5'
MEND

MACRO
M2  &P, &Q,  &U = CREG,  &V = DREG
 MOVER  &U , P
 MOVER  &V , Q
  ADD   U ,  ='15'
  ADD   V ;  ='10'
MEND
```

Calls: M1 10, 20, &B=CREG, &A=DREG
M2 100, 200, &V=AREG, &U=BREG

Algorithm in short (Pass II of Macro)

1. Create APTAB
2. Copy values from KPDTAB to APTAB
3. Copy values of posibional parameters into APTAB
4. Process keyword parameters in call. (search in KPDTAB, get 'Q')
   ⇒ put the value of that parameter in APTAB as follows:
   APTAB[#PP + Q - KPDTP + 1]

↗ either override defaults / process keyword parameters in calls

Pass I data structures:
Common for both def^n's:

MNT	name	#PP	#KP	#EV	MDTP	KPDTP	SSTP
1	EVAL	3	0	0	1	1	1
2	CLEAR	2	1	1	8	2	3

KPDTP

1	—	—
2	REG	AREG

SSTAB

1	-	6
2	-	7
3	-	11

} 1st Macro

} 2nd Macro

MDT

```
1   AIF (CP,2) EQ (P,1) (S,1)
2   MOVER AREG, (P,1)
3   SUB AREG, (P,2)
4   ADD AREG, (P,3)
5   AGO (S,2)
6   MOVER AREG, (P,3)
7   MEND
8   LCL (E,1)
9   (E,1) SET 0
10  MOVER (P,3), ='0'
11  MOVEM (P,3), (P,1)+(E,1)
12  (E,1) SET (E,1)+1
13  AIF ((E,1) NE (P,2)) (S,1)
14  MEND
```

PNTAB for EVAL

1 - P - (P,1)
2 - Q - (P,2)
3 - R - (P,3)

EVNTAB (nothing)

SSNTAB

1 - ONLY - (S,1)
2 - OVER - (S,2)

PNTAB for CLEAR

1 - X - (P,1)
2 - N - (P,2)
3 - REG - (P,3)

EVNTAB

1 - M - (E,1)

SSNTAB

1 - MORE - (S,1)

Q.1. Consider the following code segment (10M)

Unit test II

```
MACRO
    INCR &M_V, &I_V, &REG
    MOVER &REG, &M_V
    ADDS &M_V, &I_V
    MOVEM &REG, &M_V
MEND
MACRO
    ADDS &F, &S
    MOVER AREG, &F
    ADD AREG, &S
    MOVEM AREG, &S
    WRITE &S
MEND
MACRO
    SUBS &F, &S
    MOVER BREG, &F
    SUB BREG, &S
    MOVEM BREG, &S
    WRITE &S
MEND
START 200
READ N1
READ N2
ADDS N1, N2
SUBS N1, N2
INCR N1, N2, DREG
STOP
N1 DS 2
N2 DS 2
```

Show the contents of

1) MNT 2) MDT 3) ALA
Argument List Array

~~4) Expanded code~~

PU - May 2008 (10 Marks) Q.2. 2

```
START   100
SR      2,2
USING   *,15
 MACRO
      XYZ   &A
       A     1,&A
       AR    2,2
 MEND
 L     1,DI
  MACRO
        ABC    &Z
        SR     3,3
        MACRO
             DISPLAY
              XYZ   B
        MEND
        L     1,&Z
  MEND
   XYZ    B1
   SR     4,4
   ABC     B1
DI    DC      F'4'
B1    DC      F'5'
 END
```

Show the contents of

1) MNT 2) MDT 3) ALA ~~4) Expanded code~~