

**[Exercise 1]****1. 请找出程序中的错误(共 10 处?)**

```
000100  /*** QUIZ*1  PROGRAM ****/
000200  QUIZ*1 : PROC OPTION(MAIN);
000300  /*** INPUT FILE LAYOUT  ***/
000400  DCL
000500      1 IN_AREA,
000600      3 IN_1   CHAR(6),
000700      3 IN_2   FIXED DEC(7),
000800      3 IN_3   FIXED DEC(7,0),
000900      3 IN_4   FIXED DEC(7,0);
001000  /*** OUTPUT FILE LAYOUT  ***/
001100  DCL
001200      1 OUT_AREA,
001300      3 OUT_A,
001400      5 OUT_1   CHAR(6) INIT('RECORD'),
001500      5 OUT_2   FIXED DEC(3,0),
001600      7 OUT_B,
001700      5 OUT_3   FIXED DEC(11,0),
001800      5 OUT_4   FIXED DEC(11,0);
001900  /*** FILE DEFINITION ***/
002000  DCL
002100      FIN   FILE RECORD,
002200      FOUT  FILE RECORD;
002300  /*** WORK AREA  ***/
002400  DCL EOF FIXED DEC(1) INIT(0);
002500
002600  ON ENDFILE(FIN) EOF = 1 ;
002700  /*** FILE OPEN  ****/
002800  OPEN  FILE(FIN)  INPUT;
002900  OPEN  FILE(FOUT) OUTPUT;
003000  /*** MAIN PROCESS  ****
003100  READ FILE(FIN) INTO(IN_AREA);
003200  DO WHILE (EOF=0)
003300      IF IN_2 =< 100
003400          THEN DO ;
003500              OUT_3 = IN_3 * 1.25;
003600              OUT_4 = IN_4 * 1.15;
003700          ELSE DO;
003800              OUT_3 = IN_3 * 0.98;
```

```

003900      OUT_4 = IN_4 * 0.87;
004000      END;
004100      WRITE FILE(FOUT) INTO(OUT_AREA);
004200      .....
004210
004300      READ FILE(FIN) INTO(IN_AREA);
004400      END;
004500      /*** FILE CLOSE   ***/
004600      CLOSE FILE(FIN),
004700          FILE(FOUT);
004800      END QUIZ*1;

```

2. 下面的程序读入品名,单价,数量 3 个数据,输出品名和合计金额. 请完成下面的程序.

输入 ... .. 品名 : **JUICE**

单价 : **100 YUAN**

数量 : **200 BOTTLES**

处理 ... .. 合计金额=单价 \* 数量

输出 ... .. 品名 : **JUICE**

合计金额 : **20000YUAN**

MONDAI: ① \_\_\_\_\_

② W\_HINMEI ③ \_\_\_\_\_ (07),

W\_TANKA FIXED DEC (05),

W\_SURYOU ④ \_\_\_\_\_ (05),

W\_KINGAKU FIXED DEC (13);

⑤ (W\_HINMEI,W\_TANKA,⑥ \_\_\_\_\_ )

( ⑦ \_\_\_\_\_,F(05),⑧ \_\_\_\_\_);

⑨ \_\_\_\_\_;

PUT EDIT ( ⑩ \_\_\_\_\_,⑪ \_\_\_\_\_ )

(A(07),X(03),⑫ \_\_\_\_\_);

⑬ \_\_\_\_\_;

**[Exercise 2]**

**Step1: Create one Data Set names “SX01I.EV\*\*\*\*.PLI”, using the following parameters:**

Space units . . . . . TRACK

Primary quantity . . 2

Secondary quantity 2

Directory blocks . . 5

Record format . . . . FB

Record length . . . . 80

Block size . . . . . 0

**Step2: copy SX01I.PLIEDU.PLI(@BLANK) to “SX01I.EV\*\*\*\*.PLI”**

**Step3: rename SX01I.EV\*\*\*\*.PLI(@BLANK) to SAMPLE1, and input the following program**

```

/*****
/* SYSTEM NAME      :  SX01                               */
/* SUBSYSTEM NAME   :  PLIEDU                              */
/* PROGRAM          :  SAMPLE1                             */
/* EXTERNAL PROGRAM :                                     */
/* AUTHOR           :  JIAYJ                                */
/* PROGRAM WRITTEN  :  2004/02/05                           */
/* NEW HENKOH DATE  :                                     */
/* FUNCTION         :  THIS IS A SAMPLE FILE FOR EXERCISE 1 */
/*                  THIS PROGRAM READ DATA FROM INPUT FILE FIRST, */
/*                  DO SOME VERY SIMPLE COMPUTE,                */
/*                  THEN WRITE DATA TO AN OUTPUT FILE.         */
/* FILES           :                                     */
/*      NO      FILE-ID      FILE DESCRIPTION      I/O      */
/*      --      -----      -----      ---      */
/*      01      FTOUR                I      */
/*      02      FLIST                0      */
*****/

SAMPLE1 : PROC OPTIONS (MAIN);
  /*** FILE AND VARIABLE DEFINITION ***/
  DCL
    FTOUR  FILE  RECORD,
    FLIST  FILE  RECORD,

```

```

CNT_IN                FIXED DEC(5) INIT(0),
CNT_OUT               FIXED DEC(5) INIT(0),
EOF                  FIXED DEC(1) INIT(0);
/** INPUT FILE LAYOUT      */
DCL
1 IN_AREA,
  3 FILE_ID            CHAR(2),
  3 SAKUSEI_BI         FIXED DEC(7),
  3 UKETSUKE_KIKAN,
    5 KAISHI_BI        FIXED DEC(7),
    5 SYURYO_BI        FIXED DEC(7),
  3 TOUR_CODE,
    5 KAIGAI           CHAR(1),
    5 T_CODE           CHAR(2),
    5 SHUPPATSU_TSUKI  CHAR(2),
    5 JUNBAN           CHAR(1),
  3 SHUPPATSU_BI       FIXED DEC(7),
  3 TOUT_MEI          CHAR(70),
  3 HIYOH ,
    5 SAITEI_CLASS     FIXED DEC(5),
    5 HYOJUN_CLASS     FIXED DEC(5),
    5 SAIKOH_CLASS     FIXED DEC(5),
  3 MISHIYO           CHAR(27);
/** OUTPUT FILE LAYOUT    */
DCL
1 OUT_AREA,
  3 CTL                CHAR(1) INIT('0'),
  3 FILE_ID            CHAR(2),
  3 DUM01              CHAR(1) INIT(' '),
  3 TOUR_CODE,
    5 KAIGAI           CHAR(1),
    5 T_CODE           CHAR(2),
    5 DUM02            CHAR(1) INIT(' '),
    5 SHUPPATSU_TSUKI  CHAR(2),
    5 DUM03            CHAR(1) INIT(' '),
    5 JUNBAN           CHAR(1),
    5 DUM04            CHAR(1) INIT(' '),
  3 SAKUSEI_BI         PIC' (6)9' ,
  3 DUM05              CHAR(1) INIT(' '),

```

```
3 UKETSUKE_KIKAN,
  5 KAISHI_BI          PIC' (6)9' ,
  5 DUM06              CHAR(1) INIT(' '),
  5 SYURYO_BI          PIC' (6)9' ,
3 DUM07                CHAR(1) INIT(' '),
3 SHUPPATSU_BI         PIC' (6)9' ,
3 DUM08                CHAR(1) INIT(' '),
3 TOUT_MEI             CHAR(70),
3 HIYOH ,
  5 SAITEI_CLASS       PIC' ZZZ,ZZ9' ,
  5 DUM09              CHAR(1) INIT(' '),
  5 HYOJUN_CLASS       PIC' ZZZ,ZZ9' ,
  5 DUM10              CHAR(1) INIT(' '),
  5 SAIKOH_CLASS       PIC' ZZZ,ZZ9' ,
  5 DUM11              CHAR(1) INIT(' '),
3 MISHIYO              CHAR(27);

ON ENDFILE(FTOUR) EOF = 1 ;
OPEN  FILE(FTOUR) INPUT ,
      FILE(FLIST) OUTPUT;
/*** READ INPUT FILE **/
READ FILE(FTOUR) INTO(IN_AREA);

DO WHILE ( EOF = 0 );
  CNT_IN = CNT_IN + 1 ;
  OUT_AREA = IN_AREA, BY NAME;
  /*** COMPUTE ***/
  OUT_AREA.SAITEI_CLASS = IN_AREA.SAITEI_CLASS * 1000 ;
  OUT_AREA.HYOJUN_CLASS = IN_AREA.HYOJUN_CLASS * 1000 ;
  OUT_AREA.SAIKOH_CLASS = IN_AREA.SAIKOH_CLASS * 1000 ;

  /*** WRITE OUTPUT FILE ***/
  WRITE FILE(FLIST) FROM(OUT_AREA);
  CNT_OUT = CNT_OUT + 1;

  /*** READ INPUT FILE **/
  READ FILE(FTOUR) INTO(IN_AREA);
END;

/*** WRITE LOG ***/
```

```

PUT PAGE EDIT('** SAMPLE1 ASCA DATA **') (A);
PUT SKIP(2) EDIT(' INPUT =>', CNT_IN,
                ' OUTPUT=>', CNT_OUT)
                (A, X(3), P'ZZZ,ZZ9', SKIP, A, X(3), P'ZZZ,ZZ9');
CLOSE FILE(FTOUR),
        FILE(FLIST);
END SAMPLE1;
/** WRITE OUTPUT FILE **/
WRITE FILE(FLIST) FROM(OUT_AREA);
CNT_OUT = CNT_OUT + 1;

```

**Step4: create a dataset names “SX01I.EV\*\*\*\*.JCL”, using the following parameters:**

```

Space units . . . . . TRACK
Primary quantity . . 2
Secondary quantity    2
Directory blocks . . 5
Record format . . . . FB
Record length . . . . 80
Block size . . . . . 0

```

**Step5: create a dataset names “SX01I.EV\*\*\*\*.LMOD”, using the following parameters:**

```

Space units . . . . . TRACK
Primary quantity . . 2
Secondary quantity    2
Directory blocks . . 5
Record format . . . . U
Record length . . . . 0
Block size . . . . . 800

```

**Step6: copy SX01I.PLIEDU.JCL( SAMPLE1C) to SX01I.EV\*\*\*\*.JCL**  
**change the following red marked part to your owner used id(EV\*\*\*\*)**

```

000100 //JIAJCL JOB (ISSC#), ' JIAJ ' , CLASS=A, MSGCLASS=H,
000200 //          NOTIFY=JIAJ
000300 //*****
.....
007600 //SYSIN DD DUMMY
007700 // PEND
007800 //*****

```

```

007900 //STEP01 EXEC IBMZCPL, GOPGM='SAMPLE1',
008000 // SOURCE='SX01I . PLIEDU. PLI',
008100 // LMOD='SX01I . PLIEDU. LMOD'
008200 /*

```

**Step7: Submit SX01I.EV\*\*\*\*.JCL (SAMPLE1C) and resolve the errors if there are.**

**Step8: Copy SX01I . PLIEDU. SAMPLE1. FTOUR to SX01I . EV\*\*\*\*. SAMPLE1. FTOUR**

**Step9: copy SX01I.PLIEDU.JCL (SAMPLE1E) to SX01I.EV\*\*\*\*.JCL**  
**change the following red marked part to your owner used id(EV\*\*\*\*)**

```

//JIAYJ1E JOB (ISSC#), 'JIAYJ', CLASS=A, MSGCLASS=H,
// NOTIFY=JIAYJ
//*****
//*****
//STEP00 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE SX01I . PLIEDU. SAMPLE1. FLIST
/*
//*****
//* RUN STEP
//*****
//SAMPLE1 EXEC PGM=SAMPLE1
//STEPLIB DD DSN=CEE.SCEERUN, DISP=SHR
// DD DSN=SX01I . PLIEDU. LMOD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//FTOUR DD DSN=SX01I . PLIEDU. SAMPLE1. FTOUR, DISP=SHR
//FLIST DD DSN=SX01I . PLIEDU. SAMPLE1. FLIST, DISP=(NEW, CATLG),
// UNIT=SYSDA, SPACE=(TRK, (1, 1)),
// DCB=(RECFM=FB, LRECL=162, BLKSIZE=0)
/*

```

**Step10: Submit SX01I.EV\*\*\*\*.JCL (SAMPLE1E) and resolve the errors if there are.**

**Step11: Check the output to see whether the result is right according to the program**

logical. Compare SX01I.EV\*\*\*\*.SAMPLE1.FLIST with SX01I.PLI EDU.SAMPLE1.FLIST using 3.13.

**Step12: Catalog the Job Log for the exercise checking. From the SDSF;H, using XDC command.**

SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 2,457							DATA SET DISPLAYED	
COMMAND INPUT ==>							SCROLL ==> PAGE	
NP	JOBNAME	JobID	Owner	Prty	C	ODisp	Dest	Tot-Rec Tot-
	JIAYJPL	JOB00587	JIAYJ	144	H	HOLD	LOCAL	1,044
	JIAYJPL	JOB00588	JIAYJ	144	H	HOLD	LOCAL	54
XDC	JIAYJPL	JOB00589	JIAYJ	144	H	HOLD	LOCAL	1,359

**ON the next screen, input data like following:**

SDSF Open Print Data Set		
COMMAND INPUT ==>		SCROLL ==> PAGE
Data set name	==> SX01I.EV****.SAMPLE1.JOBLOG	
Member to use	==>	
Disposition	==> NEW (OLD, NEW, SHR, MOD)	
If the data set is to be created, specify the following.		
Volume serial will be used to locate existing data sets if specified.		
Management class	==>	(Blank for default management class)
Storage class	==>	(Blank for default storage class)
Volume serial	==>	(Blank for authorized default volume) *
Device type	==>	(Generic unit or device address) *
Data class	==>	(Blank for default data class)
Space units	==> BLKS	(BLKS, TRKS, CYLS, BY, KB, or MB)
Primary quantity	==> 5	(In above units)
Secondary quantity	==> 5	(In above units)
Directory blocks	==>	(Zero for sequential data set)
Record format	==> VBA	
Record length	==> 133	
Block size	==> 1330	

**Step 13: Confirm that you have the following file now:**

**SX01I.EV\*\*\*\*.PLI(SAMPLE1)** àSource Program

**SX01I.EV\*\*\*\*.JCL(SAMPLE1C)** àCompile JCL



**SX01I.EV\*\*\*\*.JCL(SAMPLE1E)    àExecute JCL**  
**SX01I.EV\*\*\*\*.LMOD(SAMPLE1)   àLoad Module**  
**SX01I.EV\*\*\*\*.SAMPLE1.FTOUR   àInput data**  
**SX01I.EV\*\*\*\*.SAMPLE1.FLIST   àOutput dataset**  
**SX01I.EV\*\*\*\*.SAMPLE1.JOBLOG   à JOB LOG**

**[Exercise 3]**

根据给定的资料编写相应的程序, 并进行测试. (基本步骤请参考 Exercise 2)

程序名:	<b>SAMPLE2</b>
编译 JCL:	<b>SAMPLE2C</b>
执行 JCL:	<b>SAMPLE2E</b>
INPUT DATASET:	<b>SX01I.EV*****.SAMPLE2.FTOUR</b>
OUTPUT DATASET	<b>SX01I.EV*****.SAMPLE2.FLIST</b> <b>SX01I.EV*****.SAMPLE2.FERROR</b>

Input: travel applying file (sorted by travel code) -> 60bytes

Applying No	Travel Code	Travel Name	Travel Price	Applying Person	Remark
C(03)	C(05)	C(10)	C(07)	C(05)	C(30)

Process:

1. Display program begin message -> "Program SAMPLE2 STARTED"
2. File Open
3. Read Input file
4. Process Input data Until input FILE end
  - a. If is a new travel code, send the summary of previous travel code to output report, and initialize the summary
  - b. Input data valid
  - c. If data is invalid, send to error report
  - d. If data is valid, compute the amount = price \* applying person
  - e. Send data to report
  - f. Sum data by travel code and total(travel number, applying person, amount)
  - g. Read next record
5. send the total summary to the out report
6. send the ASCA data to out report -> INPUT record number, error record number, and output report record number
7. close file

Input File Valid:

Applying No	Number?
Travel code	Number? Between 01001 and 02009?
Travel name	Empty?
Travel price	Number?
Applying person	Number?

## OUTPUT:

## 1. Error Report:

Length 70 bytes, add a "\*" before the error data and a space between each fields:

Applying No	Travel Code	Travel Name	Travel Price	Applying Person	Remark
*C(03)	*C(05)	*C(10)	*C(07)	*C(05)	C(30)

## 2. Travel report:

\*\*\*\* Travel Applying Status Report \*\*\*\*

AS OF 2004/01/01

Page 1

Travel Code	Travel Name	Applying No	Travel Price	Applying Person	Amount	Remark
01001	Beijing	001	1,111	10	\$11,111	Travel to beijing
		002	1,111	10	\$11,111	Travel to beijing
		003	1,111	10	\$11,111	Travel to beijing
		004	1,111	10	\$11,111	Travel to beijing
01001 total				40	\$44,444	
01002	Shanghai	001	1,111	10	\$11,111	Travel to Shanghai
		002	1,111	10	\$11,111	Travel to Shanghai
		003	1,111	10	\$11,111	Travel to Shanghai
		004	1,111	10	\$11,111	Travel to Shanghai
		005	1,111	10	\$11,111	Travel to Shanghai
		006	1,111	10	\$11,111	Travel to Shanghai
		007	1,111	10	\$11,111	Travel to Shanghai
		008	1,111	10	\$11,111	Travel to Shanghai

\*\*\*\* Travel Applying Status Report \*\*\*\*

AS OF 2004/01/01

Page 2

Travel Code	Travel Name	Applying No	Travel Price	Applying Person	Amount	Remark
		009	1,111	10	\$11,111	Travel to Shanghai
		010	1,111	10	\$11,111	Travel to Shanghai
01002 subtotal				100	\$111,110	
Total	2 travel	14 applying		140	\$155,554	

**Notes: one page 20 lines**

**[Exercise 4]**

根据给定的资料编写相应的程序, 并进行测试. (基本步骤请参考 Exercise 2)

程序名:	SAMPLE3
编译 JCL:	SAMPLE3C
执行 JCL:	SAMPLE3E
INPUT DATASET:	SX01I.EV*****.SAMPLE3.IDNM90 SX01I.EV*****.SAMPLE3.IDMM40 SX01I.EV*****.SAMPLE3.IASTCHK
OUTPUT DATASET	SX01I.EV*****.SAMPLE3.ODNM90M SX01I.EV*****.SAMPLE3.ODNM90U

**INPUT:****Inventory DB抽出済みファイル IDNM90**

KEY(UNIQUE) : 入庫M/T(INKMT)、入庫S/N(INKSN)

No	項目名	コーディング名	属性	長さ	位置	Remark
1	入庫M/T	INKMT	CH(04)	4	1	Key
2	入庫S/N	INKSN	CH(07)	7	5	Key
3	顧客番号	ICST1	CH(07)	7	12	
4	契約番号	ICNT	CH(11)	11	19	
5	物件ステータス	SEQP	CH(01)	1	30	
6	処分区分	SHOBUN	CH(01)	1	31	
7	契約メインステータス	SCNT	CH(01)	1	32	
8	契約無効ステータス	SCNTEND	CH(01)	1	33	
9	資産番号	IAST	CH(13)	13	34	
10	Load No(入庫)	ILOAD1	CH(23)	23	47	
11	承認日	DAPPR8	CH(08)	8	70	
12	承認済サイン	SAPPR	CH(01)	1	78	
13	承認者	IAPPR	CH(05)	5	79	
14	登録日	DREG8	CH(08)	8	84	
15	登録時刻	TREG	CH(06)	6	92	
16	更新日	DUPDATE8	CH(08)	8	98	
17	更新時刻	TUPDATE	CH(06)	6	106	
18	担当者[OP]	IOPERAT	CH(05)	5	112	
19	更新プログラムID	IUPDATEP	CH(08)	8	117	

**IAST重複確認用ファイル (IASTCHK)**

	I0FLE	CODING名称	項目名	属性	LOCATION
1	IASTCHK	IAST	資産番号	CH(13)	1

## 物件マスター抽出済みファイル IDMM40

	IOFLE	CODING名称	項目名	属性	LOCATION
1	IDMM40	IMM40-iMT	機種	CH(4)	1
2	IDMM40	IMM40-iSERIAL	機械番号	CH(7)	5
3	IDMM40	IMM40-iAST	資産番号	CH(13)	12
4	IDMM40	IMM40-iCST	顧客番号	CH(7)	25
5	IDMM40	IMM40-SCNT	契約ステータス	CH(1)	32
6	IDMM40	IMM40-SCNTEND	契約無効ステータス	CH(1)	33

## Process:

## A: 処理概要

1. InventoryDB 抽出済みファイル (IDNM90)、物件マスター抽出済みファイル (IDMM40) からデータを読み込む。

2. IDNM90 からの入力データの M/T の上 3 桁の値を取得する。

- ≠ '998' の時、M/T + SERIAL (全桁) + 顧客番号 (ICST1 の 上5桁) でマッチングを行う。  
マッチした時、IDMM40 の対象データの IAST が IAST 重複確認用ファイル (IASTCHK) に存在しているか確認する。

存在していれば、KEY 項目が小さいほうから再度データを入力する。

存在していなければ、IDMM40 のステータスを確認する (DMM40.SCNT=6 AND DMM40.SCNTEND=0 (リース実行中))

満たした場合、IDNM90 対象データのマッチステータス等を更新してアンマッチファイル (ODNM90U) に出力する。

満たしない場合、IDNM90 対象データのマッチステータス等を更新してマッチファイル (ODNM90M) に出力する。

マッチしなかった時、KEY 項目が小さいほうから再度データを入力する。

- '998' の時、マッチングは行わない。入力データを Inventory アンマッチファイル (ODNM90U) に出力する。上記処理を IDNM90 のデータがなくなるまで行う。

## B: 前提条件

入力ファイルは、ともに M/T>SERIAL の順に SORT されていること。

## OUTPUT:

## Inventory DB マッチファイルファイル ODNM90M

KEY(UNIQUE): 入庫M/T(INKMT)、入庫S/N(INKSN)

No	項目名	コーディング名	属性	長	位置	Remark
1	入庫M/T	INKMT	CH(04)	4	1	IDNM90
2	入庫S/N	INKSN	CH(07)	7	5	IDNM90
3	顧客番号	ICST1	CH(07)	7	12	IDNM90
4	契約番号	ICNT	CH(11)	11	19	IDNM90

5	物件ステータス	SEQP	CH(01)	1	30	IDNM90
6	処分区分	SHOBUN	CH(01)	1	31	IDNM90
7	契約メインステータス	SCNT	CH(01)	1	32	IDNM90
8	契約無効ステータス	SCNTEND	CH(01)	1	33	IDNM90
9	資産番号	IAST	CH(13)	13	34	IDMM40
10	Load №(入庫)	ILOAD1	CH(23)	23	47	IDNM90
11	承認日	DAPPR8	CH(08)	8	70	IDNM90
12	承認済サイン	SAPPR	CH(01)	1	78	"1"
13	承認者	IAPPR	CH(05)	5	79	IDNM90
14	登録日	DREG8	CH(08)	8	84	IDNM90
15	登録時刻	TREG	CH(06)	6	92	IDNM90
16	更新日	DUPDATE8	CH(08)	8	98	SYSTEM日付
17	更新時刻	TUPDATE	CH(06)	6	106	SYSTEM時刻
18	担当者[OP]	IOPERAT	CH(05)	5	112	ブランク
19	更新プログラムID	IUPDATEP	CH(08)	8	117	SAMPLE3

## Inventory DB Unmatched ODNM90U

KEY(UNIQUE) : 入庫M/T(INKMT)、入庫S/N(INKSN)

No	項目名	コーディング名	属性	長	位置	Remark
1	入庫M/T	INKMT	CH(04)	4	1	IDNM90
2	入庫S/N	INKSN	CH(07)	7	5	IDNM90
3	顧客番号	ICST1	CH(07)	7	12	IDNM90
4	契約番号	ICNT	CH(11)	11	19	IDNM90
5	物件ステータス	SEQP	CH(01)	1	30	IDNM90
6	処分区分	SHOBUN	CH(01)	1	31	IDNM90
7	契約メインステータス	SCNT	CH(01)	1	32	IDNM90
8	契約無効ステータス	SCNTEND	CH(01)	1	33	IDNM90
9	資産番号	IAST	CH(13)	13	34	IDNM90
10	Load №(入庫)	ILOAD1	CH(23)	23	47	IDNM90
11	承認日	DAPPR8	CH(08)	8	70	IDNM90
12	承認済サイン	SAPPR	CH(01)	1	78	"0"
13	承認者	IAPPR	CH(05)	5	79	IDNM90
14	登録日	DREG8	CH(08)	8	84	IDNM90
15	登録時刻	TREG	CH(06)	6	92	IDNM90
16	更新日	DUPDATE8	CH(08)	8	98	SYSTEM日付
17	更新時刻	TUPDATE	CH(06)	6	106	SYSTEM時刻
18	担当者[OP]	IOPERAT	CH(05)	5	112	ブランク
19	更新プログラムID	IUPDATEP	CH(08)	8	117	SAMPLE3