

## 画面型入出力ソフトウェア（IPF）紹介

工学部 吉田 博

### はじめに

コンピュータ利用のシステムの複雑化とコンピュータシステムの利用分野の拡大が急速に進行しているが、これとは全く無関係にコンピュータオペレーションの簡素化・入出力の簡易化に対する要求が増大してきている。例えば、数値計算における大量の入力データを間違いなく、正確に入力するにはかなりの労力と神経を消耗するのが一般である。また、既に完成しているプログラムの使用の際には、データの入力方法、順序など忘れていることが多く、調べるのに時間を浪費することもしばしばである。

コンピュータの利用分野の拡大に伴い、コンピュータおよびプログラム内容に全く無知の者がオペレーションすることも多くなってきた。

IPF（Interactive Programming Facilities）はCOBOLやFORTRAN 77、PL/Iの高級言語からTSSコマンドを呼び出したり、フルスクリーン型ディスプレイ端末に対する入出力が行えるプログラミングパッケージであり、新たなユーザニーズに対応するシステム開発での生産性向上とマンマシンインタフェースの向上を目的としたソフトウェアツールである。

IPFを利用したシステムでは、大きく分けてつぎの2つの処理が可能となる。

- (1) TSSのもとでのコマンドが応用プログラムの中で実行させることができる。たとえば、プログラムの実行中にバッチジョブをサブミットしたり、データセットを確保（消去）するコマンドを実行させることができる。
- (2) フルスクリーンディスプレイ端末への入出力アクセスを簡単に行える機能を持っている。例えば、メニュー画面によるデータの照会、画面からのデータ入力、帳票イメージの画面出力などに応用が可能である。

### 1. 例題による実習（その1）

ここでは、上記の2つの機能を含めたFORTRAN 77の例題を示し、その解説をするので、その内容をよく理解・検討されて各自のプログラムに応用されることをお勧めする。

## 問題－1

土木建設工学科の構造力学第1の成績を読み込み、データセットに書込むプログラムを作れ。  
ただし、つぎの各項目を画面の指定位置より入力し、また、画面で確認するものとする。

1. 学籍番号 (数値のみで10桁)
2. 氏名 (姓・名ともアルファベットで20字以内)
3. 試験結果  
    中間試験 (8回分で各回最大3桁の整数)  
    期末試験 (1回分で最大3桁の整数)
4. 試験のウェイト (中間試験と期末試験のウェイトをそれぞれ1桁の整数で与える)
5. 評価 (平均点を計算して表示する。順位も表示できるようにしておく)
6. メッセージ (60点以上の学生は合格と、60点未満の学生は不合格と表示)

つぎの手順で2つのプログラムを各自のデータセットにコピーして下さい。

1. KYOIKU.EXP.FORT77(IPF01)を、タイプ名がFORT77の自分のデータセット  
AB9999.XXXX.FORT77(IPF01)にコピーして下さい。これが問題－1のプログラムで  
す。内容は後程説明します。
2. KYOIKU.EXP.DATA(IPF01)を、タイプ名がDATAの自分のデータセット  
AB9999.XXXX.DATA(IPF01)に次の要領でコピーして下さい。これが問題－1のメニ  
ュー画面用のメニュー定義体と呼ばれるものです。この作り方は後程説明します。  
(1) READY 状態で PFD 2 と入力して下さい(図1)。



READY  
PFD 2

図1 EDIT 用データセットメニューの起動

(2) プロジェクト名欄に自分の課題番号AB9999を、ライブラリ名に XXXX を、タイプ名に  
DATA を、メンバー名にIPF01と入力して下さい。このとき、メンバー名IPF01は新規の  
メンバー名でなければなりません。すでに、メンバー名としてIPF01がある場合は、一度削除  
するか、別のメンバー名を選んで下さい。日本語欄には YES を入れておいて下さい。すべての  
入力が終わったなら実行キーを押して下さい。図2のような画面が現れます。

(3) コマンド欄に COPY と入れて実行キーを押して下さい。図3のような画面が現れます。

プロジェクト名欄にはコピーをとりたいデータセットの課題番号KYOIKUを、ライブラリ名に

はEXPを、タイプ名にはDATAを、メンバー名にはIPF01を入れ、さらに、行番号の有無の欄にはNOを入れて下さい。これで、実行キーを押すとノーナンバーのデータセットのコピーが終了します。

このように、メニュー画面用のメニュー定義体は、ノーナンバーでなければなりません。

```

日本語EDIT --- AB9999.XXXX.DATA(IPF01) ----- 表示欄 001 072
コマンド ==>                                     移動量 ==> CUR
***** ***** データの先頭 *****V10L20*****
*****
*****
*****
*****

```

図2 AB9999.XXXX.DATA(IPF01)のEDIT 画面

```

-----< EDIT 複写メニュー >-----
コマンド ==>
現データセット名: AB9999.XXXX.DATA(IPF01)

現データセットに複写するPFDライブラリデータセット:
プロジェクト名 ==> KYOIKU
ライブラリ名   ==> EXP      ==>      ==>      ==>
タイプ名       ==> DATA
メンバ名       ==> IPF01      ( 区分データセットの場合 )

PFDライブラリ以外のデータセット:
データセット名 ==>
ボリューム通番 ==>          ( カタログされていない場合 )

パスワード     ==>          ( パスワード保護されている場合 )

行番号範囲 ( 省略すると全レコードが複写されます )
開始行番号     ==>
最終行番号     ==>
行番号形式     ==>          ( S : 標準, C : COBOL, R : 相対 )
行番号の有無   ==> NO      ( NO : 行番号なし, YES : 行番号あり )

複写したい場合は ENTER キーを押して下さい。
複写を止める場合は END キーを押して下さい。

```

図3 KYOIKU.EXP.DATA(IPF01)の複写

3. コピーしたAB9999.XXXX.FORT77(IPF01)をPFDの2で、EDIT状態にして以下の操作をして下さい。

(1) コマンド欄に RENUM と入力してください。

- (2) 行番号 4500, 4700, 5700 のデータセット名の課題番号を自分の課題番号に変更して下さい。
- (3) 行番号 5000 のデータセット名を、いまコピーしたメニュー定義体のデータセット名 AB9999.XXXX.DATA に変更して下さい。
- (4) 行番号 5200 の文字変数 MIA の初期値の 9 文字目から 8 文字をメニュー定義体のメンバー名 IPF01 に変更して下さい。8 文字に満たない場合には、空白を補ってください。
- (5) コマンド欄に SAVE と入力し、一度内容をセーブして下さい。
- (6) コマンド欄に FORT77 JEF LIBDD と入力して下さい。
4. 図 4 に示すメニュー画面が現れますので、以下のテストをしてみてください。

==金沢大学==		==工学部==	
土木建設工学科・構造力学第一成績入力			
○学籍番号      █ ← (カーソル)			
○氏名 (アルファベット 20 字以内)			
姓			
名			
○試験結果 (3 桁以内の整数)			
中間試験			
第 1 回	:	第 2 回	
第 3 回	:	第 4 回	
第 5 回	:	第 6 回	
第 7 回	:	第 8 回	
期末試験			
○試験のウェイト (1 桁の整数)			
中間試験	:	期末試験	
○評価			
平均点	:	順位	
メッセージ=====>			
MENU.DATA		(IPF01)	

図 4 表示されるメニュー画面

- (1) カーソルが第 1 番目の入力欄の学籍番号にあることを確認して下さい。
- (2) 学籍番号を 10 桁の数値として入力して下さい。このとき、10 個の数字でなく数字以外の文字 (例えばアルファベット等) を混入してみてください。10 桁入力すると次の入力欄の氏名の姓にカーソルが移動します。実行キーはすべての入力終了するまで押さないで下さい。
- (3) 氏名の姓および名をそれぞれ 20 文字以内のアルファベットで入力して下さい。ここでも、アルファベット以外の文字 (例えば数字等) を混入してみてください。20 文字入力すると自動的に次の入力欄にカーソルが移動します。20 文字に満たない場合はタブキーを押して下さい。

- (4) 第1回～第8回の中間試験結果をそれぞれの入力欄に3桁以内の数値で入力して下さい。2桁あるいは1桁のみでも結構です。数字以外の文字（例えばアルファベット等）を混入してみてください。必ずしも8回分すべてを入力する必要はありません。入力欄の移動はすべてタブキーを使用して下さい。
- (5) 期末試験結果を3桁以内の数値で入力して下さい。2桁あるいは1桁のみでも結構です。数字以外の文字（例えばアルファベット等）を混入してみてください。
- (6) 中間試験と期末試験のウェイトを1桁の数字で入力して下さい。1個の文字を入力するとカーソルは自動的に次の入力欄に移動します。数字以外の文字（例えばアルファベット等）を入れてみてください。
- (7) ウェイトの入力が終了すると、カーソルは最初の入力欄の学籍番号へ移動します。評価の平均点および順位はデータの入力欄ではなく、後程の計算結果の出力欄ですから、ここへは入力できません。
- (8) すべての入力が終了したら、実行キーを押してください。
5. データの入力順にデータがチェックされ、誤り（例えば、数字のみの学籍番号にアルファベットが混入している等）があれば、その入力欄のデータが高輝度になり、カーソルがその位置に移動しますので、データを正しく修正して下さい。メッセージ欄にもその旨表示されます。
6. 順次、適当なデータを20組程度入力して下さい。ただし、学籍番号のみは記録しておいて下さい。後でまた使用します。

## 2. メニュー定義体

1. このメニュー画面のメニュー定義体は、図5のようになっています。メニュー定義体は、メンバーの区分データセットのメンバーでなければなりません。レコード長は任意でよく、幅方向に広いメニュー画面を作りたいときは、レコード長を大きく取る必要があります。しかし、幅方向に広いメニュー画面を作った場合、ディスプレイ画面で見られる範囲は80桁（日本語では40文字）ですから、メニュー画面のすべてを一度に見ることはできません。PF11 および PF10 キーでスクロールさせ、メニュー画面の一部分のみを見ることができます。また、上下方向に、任意の大きさでメニュー画面を作ることができます。このときも、ディスプレイ画面で見られる範囲は24行で、メニュー画面をすべて一度に見ることはできません。PF8 または PF7 キーでスクロールさせ、メニュー画面の一部分のみを見ることができます。このやり方は後に説明します。



(1) アトリビュート追加部。後で説明します。

## (2) メニュー原型部

メニュー原型部はアトリビュート文字と出力値またはパラメータから成っています。

例えば、メニュー原型部2行目の

¥==金沢大学===== 工学部

の、¥は出力のみが許されるアトリビュート文字であり、¥以後の出力値を低輝度で表示させます。

また、メニュー原型部3行目の

& 土木建設工学科・構造力学第一成績入力

の、&は出力のみが許されるアトリビュート文字であり、&以後の出力値を高輝度で表示させます。

メニュー原型部の6行目の

¥      ○学籍番号      | A                      ¥

は、¥から|の直前までのデータを低輝度で出力し、|から次の¥までは、Aと言うパラメータに|と¥で挟まれた10桁の間の値が入力されることを意味し、表示は低輝度です。ここに、|は|以後は入力可能なフィールドであり、入力値は低輝度で表示されることを示すアトリビュート文字です。アトリビュート文字|と¥で挟まれた範囲が10桁となっているのは、入力値が10桁になると、カーソルが自動的に次の入力パラメータ部に移動し、10桁以上の入力を不可能とするためです。

なお、アトリビュート文字の位置には入力も出力もできません。標準アトリビュート文字を表1に示します。

## (3) アクション定義部

メニュー原型部で導入されたパラメータの詳細な定義を行う部分です。〈ACTION〉で始まり〈END〉で終わります。

例えば、アクション定義部の1行目は

1 LA CHAR(10) INIT(¥PARAM1) CURSOR;

となっています。ここに、

1 はパラメータ番号でパラメータの数だけ1から順番に付けます。

A はパラメータ名でメニュー原型部のパラメータ名に対応しています。

CHAR(10) はパラメータの属性が文字で、その大きさが10文字であることを示しています。

INIT(¥PARAM1) は、画面に表示するとき、1番目のパラメータの値が表示されることを示します。プログラムから表示データを与える場合には必ずこれを指定する必要があります。指定しない場合は、空白となります。

CURSORは、プログラムでこのメニュー画面が呼ばれたときのカーソルの位置を示しています。最後は必ず、；（セミコロン）を付けます。

例えば、アクション定義部の17行目は

```
17 MS1 NCHAR(30) INIT(¥PARM17);
```

となっています。ここに、

17 はパラメータ番号でパラメータの数だけ1から順番に付けられています。

MS1 はパラメータ名でメニュー原型部のアトリビュート文字#に付けられたパラメータ名に対応しています。アトリビュート文字#は表1より、#以後のフィールドはパラメータ出力のみが許され、出力値は高輝度ですが、図1のメニュー定義体の最初の部分のアトリビュート追加部（<ATTR>で始まり、<END>で終る。）において、

```
#=+B ;
```

となっています。これは、#のアトリビュート属性に、表示データを1秒ごとに点滅（ブリンク）させる属性を付加することを意味しています。このように、追加可能なアトリビュート属性を表1に、また表中のアトリビュート記号の意味を表2に示します。

NCHAR(30) はパラメータの属性が日本語文字で、その大きさが日本語で30文字であることを示しています。なお、パラメータ値には日本語表示のためのシフトコードを含んではいけません（MS1=NC'学生の成績を入力して下さい'のように、引用符で囲んだ日本語文字列の前にNCが必要）。

INIT(¥PARM17)は、画面に表示するとき、17番目のパラメータの値が表示されることを示します。プログラムから表示データを与える場合には必ずこれを指定する必要があります。

最後は必ず、；（セミコロン）を付けます。

3. ここで示したメニュー定義体のパラメータの属性は、すべて文字または日本語文字でしたが、2バイトまたは4バイト精度の数値属性を指定することもできます。しかし、複雑となるので、すべて文字または日本語文字属性を指定し、文字から数値へ、数値から文字へは、プログラム内の変換サブルーチンを利用することにします。

### ★★★★ 問 題 ★★★★★

1. メニュー定義体が格納されている AB9999.XXXX.DATA(IPF01)を以下のように修正して下さい。
  - (1) メニュー画面のタイトル[土木建設工学科・構造力学第一成績入力]を反転、白ぬき表示させる。
  - (2) データの入力部分にアンダーラインを表示する。



表1 標準アトリビュート文字

標準アトリビュート文字		追 加 可 能 な ア ト リ ビ ュ ー ト 文 字														
文 字	属 性	P	N	H	L	D	A	E	B	R	O	U	V	J	S	I
⌵	入力非表示	—	○	—	—	◎	—	—	—	—	○	○	○	○	—	◎
%	入力高輝度	—	○	◎	○	—	○	○	○	○	○	○	○	○	○	◎
	入力低輝度	—	○	—	○	—	○	○	○	○	○	○	○	○	○	◎
&	出力高輝度	◎	—	◎	—	—	—	—	○	○	○	○	○	—	—	—
¥	出力低輝度	◎	—	—	—	—	—	—	○	○	○	○	○	—	—	—
#	パラメータ出力 高輝度	◎	○	◎	○	—	○	○	○	○	○	○	○	○	○	—
@	パラメータ出力 低輝度	◎	○	—	○	—	○	○	○	○	○	○	○	○	○	—

◎：標準使用    ○：追加可能なアトリビュート記号

表2 追加可能なアトリビュート記号と意味

アトリビュート 記号	意 味
A	リフレッシュ処理用のアトリビュート記号であり、リフレッシュ処理のときの更新不可を表す。
B	表示データを1秒ごとに点滅させる。ブリンク
D	非表示
E	リフレッシュ処理用のアトリビュート記号であり、リフレッシュ処理のときの消去を表す。
H	高輝度表示
I	入力フィールド
J	右詰め
K	日本語データフィールド
L	ライトペン検知フィールド（割込み）
M	端末あるいはライトペン入力によりTに変わる。
N	数字ロックフィールド
O	オーバライン表示
P	保護フィールド、出力フィールド
R	反転、白ぬき表示
T	端末あるいはライトペン入出力によりMはTとなる。
U	アンダライン表示
V	アトリビュートの位置に垂直ラインを表示する。
X	無意味、スペースでもよい。
S	ライトペン検知フィールド（選択）

### 3. メニュー画面用プログラム（その1）の説明

ここでは、メニュー画面を呼出し、入力されたデータのチェックを行い、正しくない場合は再入力を促し、すべてのデータが正しいと判断された場合に所定の処理をするプログラムを、成績入力を例に示します。

このプログラム例では、IPFの有する2つの機能について、その利用方法を示します。

```

000100 C *****
000200 C *****
000300 C *****金沢大学情報処理センター演習用プログラム*****
000400 C *****ーメニュー画面操作法(その1)ー*****
000500 C *****
000600 C *****
000700 C *****
000800 C 作成年月日:昭和62年6月16日
000900 C プログラム:吉田博
001000 C *****
001100 C 自分のデータセットにコピーを取ったら,
001200 C (1) コマンド欄に RENUM と入力し, 実行キーを押す。
001300 C (2) 行番号 4500 を自分の課題番号に変更する。
001400 C (3) 行番号 4700 を自分の課題番号に変更する。
001500 C (4) 行番号 5700 を自分の課題番号に変更する。
001600 C (5) 行番号 5000 を自分の課題番号およびコピーを取ったメニュー
001700 C 定義体のデータセット名に変更する。
001800 C (6) 行番号 5200 の9文字目から8文字をメニュー定義体のメンバー名
001900 C に変更する。8文字に満たないときは空白を補う。
002000 C 起動方法: コマンド欄に FORT77 JEF LIBDD と入力し, 実行キーを押す。
002100 C *****
002200 C
002300 LOGICAL*4 EX
002400 INTEGER*4 RC1,RC2
002500 INTEGER*4 PIA,LSIA(32),RCA
002600 INTEGER*2 MPA,PNA,CSA(2)
002700 CHARACTER*208 MDA,MDA1
002800 INTEGER*2 LSDA(13)
002900 CHARACTER*16 MIA
003000 CHARACTER*80 CMD1,CMD2,CMD3,CMD4,CMD5
003100 CHARACTER*80 MA
003200 CHARACTER*1 CWT1,CWT2
003300 CHARACTER*10 CHI/'0123456789'/
003400 CHARACTER*11 CHIN/' 0123456789'/
003500 CHARACTER*27 CHIC/' ABCDEFGHIJKLMNOPQRSTUVWXYZ'/
003600 NCHARACTER*30 MS1,MS2
003700 C ****GKNO=学籍番号;SEI=姓;MEI=名;CKN(8)=中間試験結果;
003800 C ****KMT=期末試験結果;
003900 C ****WT1=中間試験のウエイト;WT2=期末試験のウエイト
004000 C ****HKN=平均点;JNI=順位
004100 CHARACTER*10 GKNO
004200 CHARACTER*20 SEI,MEI
004300 REAL HKN
004400 INTEGER CKN(8),KMT,JNI,WT1,WT2
004500 DATA CMD1/'ALLOC F(FT60F001) DA('AB9999.WORK1.OUTLIST') MOD'/

```

```

004600 DATA CMD2/'ATTR AA BL(88) LR(88) REC(F) DS(PS)'/
004700 DATA CMD3/'ALLOC F(FT60F001) DA('AB9999.WORK1.OUTLIST') NE
004800 & SP(5,2) T CA US(AA)'/
004900 DATA CMD4/'FREE AT(AA)'/
005000 DATA CMD5/'ALLOC F(MENULIB) DA('AB9999.XXXX.DATA') SH'/
005100 DATA LSIA/32*0/
005200 DATA MIA /'MENULIB IPF01 '/
005300 DATA LSDA /24,80,1,24,1,80,0,0,0,0,0,0,0/
005400 C ***** 出力用データセットの確保と論理機番への結合 *****
005500 C
005600 C ***** 出力用データセットの有無の確認 *****
005700 INQUIRE(FILE='AB9999.WORK1.OUTLIST',EXIST=EX)
005800 IF (EX) GOTO 201
005900 C ***** 出力用データセットが確保されていない場合 *****
006000 CALL IPFCMD(RC1,RC2,CMD2,80)
006100 IF (RC1.NE.0) THEN
006200 PRINT *, '*** ATT ERROR ***',
006300 & 'RC1=',RC1,' RC2=',RC2
006400 GOTO 9999
006500 ENDIF
006600 CALL IPFCMD(RC1,RC2,CMD3,80)
006700 IF (RC1.NE.0) THEN
006800 PRINT *, '*** ALLOC ERROR ***',
006900 & 'RC1=',RC1,' RC2=',RC2
007000 GOTO 9999
007100 ENDIF
007200 CALL IPFCMD(RC1,RC2,CMD4,80)
007300 IF (RC1.NE.0) THEN
007400 PRINT *, '*** ATT FREE ERROR ***',
007500 & 'RC1=',RC1,' RC2=',RC2
007600 GOTO 9999
007700 ENDIF
007800 GOTO 202
007900 C ***** 出力用データセットが確保されている場合 *****
008000 201 CALL IPFCMD(RC1,RC2,CMD1,80)
008100 IF (RC1.NE.0) THEN
008200 PRINT *, '*** ALLOC ERROR ***',
008300 & 'RC1=',RC1,' RC2=',RC2
008400 GOTO 9999
008500 ENDIF
008600 202 OPEN (60)
008700 CALL IPFCMD(RC1,RC2,CMD5,80)
008800 C ** IPF01 の画面のオープン *****
008900 PIA=0
009000 CALL IPFOVS(PIA,LSIA,LSDA,RCA)
009100 IF (RCA.NE.0) PRINT *, '*** IPFOVS ERROR *** RCA=', RCA
009200 C *****
009300 C
009400 C
009500 MS1=NC'学生の成績を入力して下さい。'
009600 CWT1=' '
009700 CWT2=' '
009800 1000 MDA=' '
009900 MDA(78:78)=CWT1
010000 MDA(79:79)=CWT2

```

```

010100      CSA(1)=1
010200      CSA(2)=1
010300      MS2=NC'終了はP F 3キーまたはP F 1 5キーを押して下さい。'
010400      MDA(89:148)=MS1
010500      MDA(149:208)=MS2
010600      CALL IPFMIO(LSIA,MIA,MDA,CSA,RCA)
010700      IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 3000
010800      IF (RCA.NE.0) THEN
010900          PRINT *, '*** IPFMIO ERROR *** RCA=', RCA
011000          GOTO 3000
011100      ENDIF
011200 C    ***学籍番号のデータチェックと文字変数の代入*****
011300      PNA=0
011400 2001 DO 10 I=1,10
011500          ID=INDEX(CHI,MDA(I:I))
011600          IF (ID.EQ.0) THEN
011700              MS1=NC'学籍番号に数値以外の文字を検出'
011800              PNA=1
011900              CSA(1)=1
012000              CSA(2)=1
012100              GOTO 2101
012200          ENDIF
012300 10 CONTINUE
012400      GKNO=MDA(1:10)
012500 C    ***氏名の姓のデータチェックと文字変数の代入*****
012600      DO 20 I=11,30
012700          ID=INDEX(CHIC,MDA(I:I))
012800          IF (ID.EQ.0) THEN
012900              MS1=NC'氏名の姓にアルファベット以外の文字を検出'
013000              PNA=2
013100              CSA(1)=2
013200              CSA(2)=1
013300              GOTO 2101
013400          ENDIF
013500 20 CONTINUE
013600      SEI=MDA(11:30)
013700 C    ***氏名の名のデータチェックと文字変数の代入*****
013800      DO 30 I=31,50
013900          ID=INDEX(CHIC,MDA(I:I))
014000          IF (ID.EQ.0) THEN
014100              MS1=NC'氏名の名にアルファベット以外の文字を検出'
014200              PNA=3
014300              CSA(1)=3
014400              CSA(2)=1
014500              GOTO 2101
014600          ENDIF
014700 30 CONTINUE
014800      MEI=MDA(31:50)
014900 C    ***中間試験結果のデータチェックと文字から整数への変換*****
015000      DO 40 I=1,8
015100          II=51+(I-1)*3
015200          IF (MDA(II:II+2).EQ.' ') THEN
015300              CKN(I)=999
015400              GOTO 40
015500      ENDIF

```

```

015600      DO 50 J=0,2
015700          ID=INDEX(CHIN,MDA(II+J:II+J))
015800          IF (ID.EQ.0) THEN
015900              MS1=NC'中間試験結果に空白, 数値以外の文字を検出'
016000              PNA=I+3
016100              CSA(1)=I+3
016200              CSA(2)=J+1
016300              GOTO 2101
016400          ENDIF
016500      50  CONTINUE
016600          CALL Ciset(MDA(II:II+2),CKN(I),MA,ILL)
016700          IF (ILL.NE.0) THEN
016800              PNA=I+3
016900              CSA(1)=I+3
017000              CSA(2)=1
017100              MS1=MA
017200              GOTO 2101
017300          ENDIF
017400      40  CONTINUE
017500 C  ***期末試験結果のデータチェックと文字から整数への変換*****
017600          DO 60 J=0,2
017700              ID=INDEX(CHIN,MDA(75+J:75+J))
017800              IF (ID.EQ.0) THEN
017900                  MS1=NC'期末試験結果に空白, 数値以外の文字を検出'
018000                  PNA=12
018100                  CSA(1)=12
018200                  CSA(2)=J+1
018300                  GOTO 2101
018400              ENDIF
018500      60  CONTINUE
018600          IF (MDA(75:77).EQ.' ') THEN
018700              KMT=999
018800              GOTO 70
018900          ENDIF
019000          CALL Ciset(MDA(75:77),KMT,MA,ILL)
019100          IF (ILL.NE.0) THEN
019200              PNA=12
019300              CSA(1)=12
019400              CSA(2)=1
019500              GOTO 2101
019600          ENDIF
019700 C  ***中間試験のウエイトのチェックと文字から整数への変換*****
019800      70  ID=INDEX(CHI,MDA(78:78))
019900          IF (ID.EQ.0) THEN
020000              MS1=NC'中間試験のウエイトに数値以外の文字を検出'
020100              PNA=13
020200              CSA(1)=13
020300              CSA(2)=1
020400              GOTO 2101
020500          ENDIF
020600          CWT1=MDA(78:78)
020700          CALL Ciset(MDA(78:78),WT1,MA,ILL)
020800          IF (ILL.NE.0) THEN
020900              MS1=NC'中間試験のウエイトに数値以外の文字を検出'
021000              PNA=13

```

```

021100          CSA(1)=13
021200          CSA(2)=1
021300          GOTO 2101
021400        ENDIF
021500 C  ***期末試験のウエイトのチェックと文字から整数への変換*****
021600          ID=INDEX(CHI,MDA(79:79))
021700          IF (ID.EQ.0) THEN
021800            MS1=NC'中間試験のウエイトに数値以外の文字を検出'
021900            PNA=14
022000            CSA(1)=14
022100            CSA(2)=1
022200            GOTO 2101
022300          ENDIF
022400          CWT2=MDA(79:79)
022500          CALL C1SET(MDA(79:79),WT2,MA,ILL)
022600          IF (ILL.NE.0) THEN
022700            MS1=NC'期末試験のウエイトに数値以外の文字を検出'
022800            PNA=13
022900            CSA(1)=13
023000            CSA(2)=1
023100            GOTO 2101
023200          ENDIF
023300          GOTO 80
023400 2101 MDA(89:148)=MS1
023500          MDA(149:208)=MS2
023600          MPA=1
023700          MA='指摘した箇所を修正して、再入力して下さい。'
023800          CALL IPPMRR(LSIA,MA,MPA,MDA,PNA,CSA,RCA)
023900          IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 3000
024000          IF (RCA.NE.0) THEN
024100            PRINT *, '*** IPPMRR ERROR *** RCA=', RCA
024200            GOTO 3000
024300          ENDIF
024400          GOTO 2001
024500 80  TOTAL=0.0
024600          IN=0
024700          DO 90 I=1,8
024800            IF (CKN(I).EQ.999) GOTO 90
024900            TOTAL=TOTAL+FLOAT(CKN(I))
025000            IN=IN+1
025100 90  CONTINUE
025200 C  *****すべての成績が入力されていない場合の処理*****
025300          IF (IN.EQ.0.AND.KMT.EQ.999) THEN
025400            MS1=NC'成績が入力されていません。'
025500            CSA(1)=4
025600            CSA(2)=1
025700            GOTO 2101
025800          ENDIF
025900 C  *****中間試験の成績がすべて入力されていない場合の処理*****
026000          IF (IN.EQ.0) THEN
026100            IN=1
026200            WT1=0
026210            WT2=1
026300          ENDIF
026400 C  *****期末試験の成績が入力されていない場合の処理*****

```

```

026500      IF (KMT.EQ.999) THEN
026600          KMT=0
026700          WT1=1
026800          WT2=0
026900      ENDIF
027000 C *****平均値の計算と結果の表示*****
027100          HKN=TOTAL*FLOAT(WT1)/FLOAT(IN)+FLOAT(KMT)*FLOAT(WT2)
027200          HKN=HKN/(FLOAT(WT1+WT2))
027300          IF (HKN.LT.60.0) MS1=NC'この学生は不合格'
027400          IF (HKN.GE.60.0) MS1=NC'この学生は合格'
027500          CALL RCSET(HKN,MDA(80:85),'(F6.2)')
027600          MS2=NC'次の学生のデータ入力は、P F 3 キーを押して下さい。'
027700          MDA(89:148)=MS1
027800          MDA(149:208)=MS2
027900          MDA1=' '
028000          MDA1=MDA
028100 2102 CALL IPFMRR(LSIA,MA,MPA,MDA,PNA,CSA,RCA)
028200          IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 3100
028300          IF (RCA.NE.0) THEN
028400              PRINT *, '*** IPFMRR ERROR *** RCA=', RCA
028500              GOTO 3000
028600          ENDIF
028700          IF (MDA.NE.MDA1) GOTO 2101
028800 3100 WRITE(60,'(A88)') MDA(1:88)
028900          MS1=NC'次の学生のデータを入力して下さい。'
029000          GOTO 1000
029100 C ** IPF01 の画面のクローズ *****
029200 3000 CALL IPFCVS(LSIA,RCA)
029300          IF (RCA.NE.0) PRINT *, '*** IPFCVS ERROR *** RCA=', RCA
029800 C ***** 出力データセットのクローズ処理 *****
030000 9999 STOP
030100      END
030500 C
030600 C      文字変数から整変数（4バイト）への置換サブルーチン
030700 C      IVR4=整変数名; CHR=文字変数; ILL=0のとき正常値を返す
030800 C      ILL=8: 数字または空白以外のものを入力した
030900 C      ILL=16: 数字と数字の間に空白がある
031000 C
031100      SUBROUTINE Ciset(CHR,IVR4,MA,ILL)
031200          INTEGER*4 IVR4
031300          CHARACTER*11 CHI/' 0123456789'/
031400          CHARACTER*80 MA
031500          CHARACTER*(*) CHR
031600          CHARACTER*4  FMT/'(I )'/
031700          IC=LEN(CHR)
031800          ILL=0
031900 C      不正文字の検出
032000          IX=0
032100          DO 10 I=1,IC
032200              ID=INDEX(CHI,CHR(I:I))
032300              IF (ID.EQ.0) THEN
032400 C          空白、数値以外の文字を検出
032500              MA=NC'空白および数値以外の文字が含まれています。'
032600              ILL=8
032700              RETURN

```

```

032800      ENDIF
032900      IF (IX,EQ.0.AND.ID,NE.1) IX=1
033000      IF (IX,EQ.1.AND.ID,EQ.1) IX=2
033100      IF (IX,EQ.2.AND.ID,NE.1) THEN
033200 C      数値と数値の間に空白を検出
033300      MA=NC'数値と数値の間に空白が含まれています. '
033400          ILL=16
033500          RETURN
033600      ENDIF
033700 10 CONTINUE
033800      IF (IX,EQ.0) THEN
033900 C      総て空白を検出
034000      MA=NC'数値を入れて下さい. '
034100          ILL=32
034200          RETURN
034300      ENDIF
034400 C
034500 C      数値の桁数および位置の調査
034600 C      II=桁数; I1=数値の開始位置
034700 C
034800          II=0
034900          I1=0
035000          DO 30 I=1,IC
035100              ID=INDEX(CHI,CHR(I:I))
035200              IF (I1,EQ.0.AND.ID,NE.1) I1=I
035300              IF (ID,NE.1) II=II+1
035400              IF (II,NE.0.AND.ID,EQ.1) GOTO 20
035500 30 CONTINUE
035600 20 WRITE(FMT(3:3),'(I1)') II
035700      READ(CHR(I1:I1+II-1),FMT) IVR4
035800      RETURN
035900      END
036000 C
036100 C      文字変数から実変数 (4 バイト) への置換サブルーチン
036200 C      RVR4=実変数名; CHR=文字変数; ILL=0のとき正常値を返す
036300 C      ILL=8: 数字または空白以外のものを入力した
036400 C      ILL=16: 数字と数字の間に空白がある
036500 C
036600      SUBROUTINE CRSET(CHR,RVR4,MA,ILL)
036700      REAL*4 RVR4
036800      CHARACTER*12 CHI/' .0123456789'/
036900      CHARACTER*80 MA
037000      CHARACTER*(*) CHR
037100      CHARACTER*8 FMT/'(F00.00)'/
037200      IC=LEN(CHR)
037300      ILL=0
037400 C      不正文字の検出
037500      IX=0
037600      IP=0
037700      DO 10 I=1,IC
037800          ID=INDEX(CHI,CHR(I:I))
037900          IF (ID,EQ.0) THEN
038000 C      空白, 数値, 小数点以外の文字を検出
038100      MA=NC'空白, 数値および小数点以外の文字が含まれています.
038200          ILL=8

```



```

038300      RETURN
038400      ENDIF
038500      IF (IX.EQ.0.AND.ID.NE.1) IX=1
038600      IF (IX.EQ.1.AND.ID.EQ.1) IX=2
038700      IF (IX.EQ.2.AND.ID.NE.1) THEN
038800 C      数値と数値の間に空白を検出
038900      MA=NC'数値と数値の間に空白が含まれています。 '
039000      ILL=16
039100      RETURN
039200      ENDIF
039300      IF (ID.EQ.2) IP=IP+1
039400 10 CONTINUE
039500      IF (IX.EQ.0) THEN
039600 C      総て空白を検出
039700      MA=NC'数値を入れて下さい。 '
039800      ILL=32
039900      RETURN
040000      ENDIF
040100      IF (IP.GT.1) THEN
040200 C      小数点が2個以上ある
040300      MA=NC'小数点が2個以上あります。 '
040400      ILL=64
040500      RETURN
040600      ENDIF
040700 C
040800 C      数値の桁数および位置の調査
040900 C      II=桁数; I1=数値の開始位置; IJ=小数点位置
041000 C
041100      II=0
041200      IJ=0
041300      I1=0
041400      DO 30 I=1,IC
041500      ID=INDEX(CH1,CHR(I:I))
041600      IF (I1.EQ.0.AND.ID.NE.1) I1=I
041700      IF (ID.EQ.2) IJ=I
041800      IF (ID.NE.1) II=II+1
041900      IF (II.NE.0.AND.ID.EQ.1) GOTO 20
042000 30 CONTINUE
042100      IF (IJ.EQ.0) IJ=I1+II
042200 20 WRITE(FMT(3:4),'(I2)') II
042300      WRITE(FMT(6:7),'(I2)') II-IJ+I1-1
042400      READ(CHR(I1:I1+II-1),FMT) RVR4
042500      RETURN
042600      END
042700 C
042800 C      整変数(4バイト)から文字変数への置換サブルーチン
042900 C      IVR=整変数名; CHR=文字変数;
043000 C
043100      SUBROUTINE ICSET(IVR4,CHR,FMT)
043200      INTEGER*4 IVR4
043300      REAL*4 RVR4
043400      CHARACTER*(*) CHR,FMT
043500      WRITE(CHR,FMT) IVR4
043600      RETURN
043700 C

```

```

043800 C      実変数（４バイト）から文字変数への置換サブルーチン
043900 C      RVR=実変数名；CHR=文字変数；
044000 C
044100      ENTRY      RCSET(RVR4,CHR,FMT)
044200      WRITE(CHR,FMT) RVR4
044300      RETURN
044400      END

```

まず、最初に本プログラムで使用するサブルーチンについて説明します。

#### 1. 文字変数から整変数（４バイト）への置換サブルーチン（行番号31100-35900）

SUBROUTINE Ciset (CHR,IVR4,MA,ILL)

仮引数の説明

- (1) CHR : 整数に変換しようとする文字変数
- (2) IVIR : 文字から変換される整変数（４バイト）
- (3) MA : 不正文字が検出されたときのメッセージ用文字変数
- (4) ILL : エラーがあったときの復帰コード（ILL=0 のとき正常復帰）

#### 2. 文字変数から実変数（４バイト）への置換サブルーチン（行番号36600-42600）

SUBROUTINE CRSET (CHR,RVR4,MA,ILL)

仮引数の説明

- (1) CHR : 実数に変換しようとする文字変数
- (2) RVIR : 文字から変換される実変数（４バイト）
- (3) MA : 不正文字が検出されたときのメッセージ用文字変数
- (4) ILL : エラーがあったときの復帰コード（ILL=0 のとき正常復帰）

#### 3. 整変数（４バイト）から文字変数へ、実変数（４バイト）から文字変数への置換サブルーチン（行番号43100-44400）

SUBROUTINE ICSET (IVR4,CHR,FMT)

仮引数の説明

- (1) IVR4 : 文字に変換される整変数（４バイト）
- (2) CHR : 変換しようとする文字変数
- (3) FMT : 変換のための書式用文字変数

ENTRY RCSET (RVR4,CHR,FMT)

仮引数の説明

- (1) RVR4 : 文字に変換される実変数（４バイト）
- (2) CHR : 変換しようとする文字変数
- (3) FMT : 変換のための書式用文字変数

メインのプログラムを順を追って説明します。

2300 : EX は4バイトの論理変数の宣言です。後に説明します。

2400-2500 : RC1, RC2, PIA, RCA は4バイトの整数, LSIA は32の要素を有する整数(4バイト)の配列の宣言です。後に説明します。

2600 : MPA, PNA は2バイトの整数, CSA は2つの要素を有する整数(2バイト)の配列の宣言です。後に説明します。

2700 : 変数 MDA は208の長さを有する文字変数であることの宣言です。208文字は、メニュー定義体で使したすべての文字変数の長さの和です。後の使用時に説明します。

2800 : LSDA は13の要素を有する整数(2バイト)の配列であることの宣言です。後に説明します。

2900 : MIA は長さが16文字の文字変数であることの宣言です。後に説明します。

3000 : CMD1, CMD2, CMD3, CMD4, CMD5 は、長さが最大80文字の文字変数であることの宣言です。後に初期値が与えられます。

3100 : MA は、長さが最大80文字の文字変数であることの宣言です。後の使用時に説明します。

3200 : CWT1, CWT2 はプログラム中で一時的に使用する長さが1文字の文字変数の宣言です。

3300 : CHI は0~9迄の数字からなる長さ10文字の文字列であることの宣言です。

3400 : CHIN は空白と0~9迄の数字からなる長さ11文字の文字列であることの宣言です。

3500 : CHIC は空白とA~Zまでのアルファベットからなる長さ27文字の文字列であることの宣言です。

3600 : MS1, MS2 は長さが30文字の日本語文字変数であることの宣言です。

4100 : GKNO は学籍番号を入れる長さ10文字の文字変数であることの宣言です。

4200 : SEI, MEI は氏名の姓と名を入れる長さ20文字の文字変数であることの宣言です。

4300 : HKN は中間試験結果と期末試験結果にウェイトを考慮した平均値を入れる実数であることの宣言です。

4400 : CKN(8), KMT, JN1, WT1, WT2 は、それぞれ、8回分の中間試験結果, 期末試験結果, 順位(本プログラムでは使用しない。), 中間試験のウェイト, 期末試験のウェイトを入れる整数であることの宣言です。

4500-5000 : CMD1 ~ CMD5 にTSSコマンドを初期値として代入しています。

5100 : 32の要素を有する配列 LSIA に初期値として、すべての要素に0を入れています。プログラムの途中で LSIA の内容の変更は禁止されています。

5200 : 16文字の長さを有する MIA の始めの8文字には、CMD5 で割り付けたメニュー定義体のデータセットに対するファイル名 MENULIB と1つの空白を、後半の8文字にはメニュー定義体のメンバー名を入れます。メンバー名が8文字に満たない場合には空白を補って

下さい。

5300 : 2 バイトの整変数の 13 要素からなる配列 LSDA の各要素には次の数値を代入します。

LSDA (1) : メニュー定義体のメニュー定義部の行の大きさ (ここでは、図 4 のように、1 画面分 24 行となっています)。

LSDA (2) : メニュー定義体の列の大きさ (ここでは、1 画面分 80 列となっています)。

LSDA (3) : 画面のどの部分に表示させるかの行の上限 (ここでは 1 行目から)。

LSDA (4) : 画面のどの部分に表示させるかの行の下限 (ここでは 24 行目まで)。

LSDA (5) : 画面のどの部分に表示させるかの列の上限 (ここでは 1 列目から)。

LSDA (6) : 画面のどの部分に表示させるかの列の下限 (ここでは 80 列目まで)。

LSDA (7) : メニュー定義体のメニュー定義部の行の大きさが大き過ぎ、指定した範囲内に表示できないとき、PF7 または PF8 キーによりスクロールさせるときのスクロール行数

LSDA (8) : メニュー定義体のメニュー定義部の列の大きさが大き過ぎ、指定した範囲内に表示できないとき、PF10 または PF11 キーによりスクロールさせるときのスクロール列数

LSDA (9) : 画面を行方向にスクロールさせるとき、画面送りを禁止する第 1 行からの行数

LSDA (10) : 画面を列方向にスクロールさせるとき、画面送りを禁止する第 1 列からの列数

LSDA (11) : メニュー定義体のメニュー定義部と画面との行方向のずれ (通常は 0)

LSDA (12) : メニュー定義体のメニュー定義部と画面との列方向のずれ (通常は 0)

LSDA (13) : 端末種別 (通常は 0)

5700-5800 : 出力用データセットの有無の確認。論理変数 EX が真ならデータセットは存在し、偽ならデータセットは存在しません。

6000-7800 : 出力用データセットが確保されていない場合あらたに確保し、出力用データセットを論理機番 60 へ結合します。CALL IPFCMD (RC1, RC2, CMD2, 80) は IPF プログラミングパッケージによる TSS コマンドの実行サブルーチンの呼出で、RC1 は IPFCMD サブルーチンの復帰コードで RC1=0 なら正常復帰です。RC2 はコマンドの復帰コードです。CMD2 はコマンドを記述した文字列で行番号 4600 で与えられています。80 は CMD2 の文字列の長さです。このようにして、TSS コマンド CMD2, CMD3, CMD4 を次々に実行することができます。なお、復帰コードが 0 でない場合の処理方法についても理解する必要があります。

8000-8500 : 出力用データセットがすでに確保されている場合、出力用データセットを論理機番 60 へ結合します。

8600 : 論理機番 60 のオープン

8700 : メニュー定義体が格納されているデータセットのファイル MENULIB への結合

8900-9100 : メニュー画面のオープン処理画面のオープンに際し、PIA は必ず 0 とします。  
これ以後 PIA の内容の変更をしてはなりません。

CALL IPFOVS (PIA, LSIA, LSDA, RCA)

の LSIA は画面のオープンに際し、必ず 0 を入れておきます (行番号 5100 参照)。

LSDA は画面を制御する情報が入っている配列で、行番号 5300 で与えられています。

RCA は復帰コードであり、0 のとき正常復帰です。なお、復帰コードが 0 でない場合の処理方法についても理解する必要があります。

9500-11100 : メニューを用いた入出力処理

CALL IPFMIO (LSIA, MIA, MDA, CSA, RCA)

で、メニュー定義体で定義した画面が表示され、画面からの入出力が可能となります。

LSIA : IPFOVS で使用されたものをそのまま使用します。表示しようとする画面の識別に用いられます。

MIA : 使用するメニュー定義体が格納されているデータセットのファイル名とメンバー名がそれぞれ 8 バイトの領域に指定されています (行番号 5200 参照)。

MDA : 使用するメニュー定義体のパラメータ指定に対応して用意された領域であり、実際の転送データの格納領域です。本プログラムと引用するメニュー定義体との関係は図 6 の通りである。

CSA : メニュー画面上のカーソル設定をプログラム側から行うための 2 バイト整数の配列 (行番号 2600 参照) で、CSA (1) はカーソルを設定するメニュー定義体のパラメータ番号、CSA (2) はそのパラメータ内のオフセット (最初の文字から何番目かを示す数値) である。CSA によるカーソルの設定はメニュー定義体のアクション定義部の指定より優先する。

図 6 において、最上段はメニュー定義体の <ACTION> 部におけるパラメータ名であり、次段は MDA のパラメータの最初からのバイト数です。枠内は MDA (パラメータ) の内容であり、枠下の段は各パラメータの占めるバイト数です。最下段はプログラムにおける変数名です。このように、MDA は使用するメニュー定義体のパラメータに対応して応用プログラム側で用意し、IPF サブルーチンに引き渡す文字データの領域です。

RCA : メニューサブルーチン IPFMIO 実行後、その復帰コードが設定される領域  
(整数 4 バイト; RCA=0 のとき正常終了)

MS1=NC ' ' および MS2=NC ' ' は MS1, MS2 の初期化であり、引用符の間には日本語の空白に、MDA= ' ' は、MDA の初期化で最初に全領域を空白にしておく必要があります。

メニュー定義体 先頭からの番号	A		B		C	
	1	10 11	30 31	50		
内容	学籍番号		氏名 (姓)		氏名 (名)	
確保領域 プログラム	10 GKNO		20 SEI		20 MEI	
CH1~CH8		KMT		L		M
51	74 75	77 78	79			
中間試験 8 回分 (3×8=24)		期末試験		中間ウエイト		期末ウエイト
24 CKN (8) (整数)		3 KMT (整数)		1 WT1 (整数)		1 WT2 (整数)
N		JNI		MS1		MS2
80	85 86	88 89	148 149	208		
平均点		順位		メッセージ (1)		メッセージ (2)
6 HKN (実数)		3 JNI (実数)		60 (日本語 30 文字) MS1 (日本語)		60 (日本語 30 文字) MS2 (日本語)

図6 MDA の内容

MDA (89:148)=MS1およびMDA (149:208)=MS2は MDA の 89 バイト目から 148 バイト目まではメッセージ (1) (MS1) の領域であり、149 バイト目から 208 バイト目までがメッセージ (2) (MS2) の領域ですので初期値を代入しています。

IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 3000 は、もし、PF3 キーまたは PF15 キーが押されたとき、RCA に 3 または 15 が復帰コードとして代入されます。したがって、PF3 キーまたは PF15 キーが押されたとき、終了処理をします。

#### 11300-12400 : 学籍番号のデータチェックと文字変数の代入

学籍番号は MDA の 1 バイト目から 10 バイト目までに格納されているので、

ID=INDEX (CHI,MDA (I:I))

において、1 文字ごとに数字かどうかを調べています。CHI は行番号 3300 で、0~9 までの数字からなる文字列で、INDEX (CHI,MDA (I:I)) は 文字MDA (I:I) が文字列 CHI の何番目にあるかを調べる組み込み関数です。該当する文字が CHI に無い場合には 0 が返されます。

MS1=NC ' 学籍番号に数値以外の文字を検出 '

は、メニュー画面のメッセージ欄に表示するメッセージです。

PNA=1 は、メニュー定義体のアクション部の第 1 パラメータ、すなわち、学籍番号を高輝度表示することの指定です。

CSA(1)=1 は、カーソルを第1パラメータに位置付けし、  
CAS(2)=I は、カーソルをパラメータのI文字目に位置付けることを指定しています。  
GKNO=MDA(1:10) は、MDAの1バイト目から10バイト目までの文字を GKNO  
に代入しています。

学籍番号に数値以外のものが含まれていると、文番号 2101 へ行きます。

12600-13600 : 氏名の姓のデータチェックと文字変数の代入

氏名の姓がすべてアルファベットまたは空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字変数の代入を行っています。

13800-14800 : 氏名の名のデータチェックと文字変数の代入

氏名の名がすべてアルファベットまたは空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字変数の代入を行っています。

15000-17400 : 中間試験結果のデータチェックと文字から整数への変換を行っています。中間

試験結果がすべて数字または空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字から整数への変換を行っています。

17600-19600 : 期末試験結果のデータチェックと文字から整数への変換を行っています。期末

試験結果がすべて数字または空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字から整数への変換を行っています。

19800-21400 : 中間試験のウエイトのチェックと文字から整数への変換

中間試験のウエイトが数字または空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字から整数への変換を行っています。

21600-23200 : 期末試験のウエイトのチェックと文字から整数への変換

期末試験のウエイトが数字または空白であるかどうかのチェックを行い、そうでない場合にはエラー処理を、満足しているときは文字から整数への変換を行っています。

23400-24300 : 修正処理のためのメニュー画面上のメッセージの表示と特定パラメータの再入力を促す。

CALL IPFMRR(LSIA,MA,MPA,MDA,PNA,CSA,RCA)

は、IPFMIOサブルーチン呼出し後に、メニュー画面上のメッセージの表示と特定パラメータの再入力を促す処理を行う場合に呼出されます。

LSIA : IPFOVSで使用されたものをそのまま使用します。表示しようとする画面の識別に用いられます。

MA : MS1,MS2とは別に、メッセージとして出力するデータを入れます。ここでは、  
MA= ' 指摘した箇所を修正して、再入力して下さい。 ' のように、日本語コードを含めておく必要があります (MA= ' 指摘した箇所を修正して、再入力して下さい。 ' のように、日本語文字列の前に NC を必要としません) 。

MPA : MAのメッセージを出力しようとする行位置を指定します。ここでは1行目が指定されています。

MDA : IPFMIOで用いたものと同じです。

PNA : 輝度を変更するパラメータ番号を格納する2バイトの領域です。ここでは、入力データに誤りがあったパラメータ番号の輝度1つのみを変更しています。また、たとえばパラメータ番号1とパラメータ番号3の2つの輝度を変更したい場合には、

INTEGER\*2 PNA1,PNA2

.....

PNA1=1

PNA2=3

CALL IPFMRR(LSIA,MA,MPA,MDA,PNA1,PNA2,CSA,RCA)

のようにすればよい。

CSA : IPFMIOと同じ

RCA : IPFMIOと同じ

メッセージ欄に表示するメッセージは、行番号23400と23500でMDAに格納して用意されています。

24500-25100 : 中間試験の回数と合計点の計算

27100-27400 : ウェイトを考慮した平均点の計算と、メッセージの準備

27500 : 平均点 HKN の実数の文字データへの変換

27600-28700 : 平均点の表示と再修正処理が必要な場合の再入力を促します。

28800-29000 : 出力用データセットへの書込みと、次のデータ入力のための準備

29200-29300 : メニュー画面のクローズ処理

#### 4. 例題による実習 (その2)

##### 問題-2

土木建設工学科の構造力学第一の成績が格納されているデータセットから指定した学籍番号の学生の成績を取り出し、画面に表示せよ。このとき、学籍番号、氏名、中間試験結果、期末試験結果およびそれらのウェイトが、表示画面上で変更できるようにせよ。

1. KYOIKU.EXP.FORT77(IPF02)を、タイプ名がFORT77の自分のデータセットAB9999.XXXX.FORT77(IPF02)にコピーして下さい。これが問題-2のプログラムです。内容は後程説明します。



2. コピーしたデータセットAB9999.XXXX.FORT77 (IPF02)を PFD の2で、EDIT 状態にして以下の操作をして下さい。

3. コマンド欄にFORT77 JEF LIBDDと入力して下さい。

- (1) カーソルが第1番目の入力欄の学籍番号にあることを確認して下さい。
- (2) 学籍番号欄に先に入力した学生の学籍番号を入れて、実行キーを押して下さい。
- (3) 先程入力した成績が表示されることを確認して下さい。
- (4) 学籍番号、氏名、成績等の変更ができるかどうか試して下さい。変更した結果が正しく格納されたかどうか、もう一度学籍番号を入力し、確認して下さい。
- (5) つぎに、先に入力されていない学籍番号を入力してみてください。

## 5. メニュー画面用プログラム（その2）の説明

```

000100 C ****
000200 C ****
000300 C ****金沢大学情報処理センター演習用プログラム****
000400 C ****          -メニュー画面操作法(その2)-          ****
000500 C ****
000600 C ****
000700 C ****
000800 C ****作成年月日:昭和62年6月16日****
000900 C ****プログラム:吉田博****
001000 C ****
001100 C 自分のデータセットにコピーを取ったら,
001200 C (1) コマンド欄に RENUM と入力し, 実行キーを押す。
001300 C (2) 行番号 4500 を自分の課題番号に変更する。
001400 C (3) 行番号 4600 を自分の課題番号に変更する。
001500 C (4) 行番号 5400 を自分の課題番号に変更する。
001600 C (5) 行番号 4700 を自分の課題番号およびコピーを取ったメニュー
001700 C     定義体のデータセット名に変更する。
001800 C (6) 行番号 4900 の9文字目から8文字をメニュー定義体のメンバー名
001900 C     に変更する。8文字に満たないときは空白を補う。

```

```

002000 C 起動方法：コマンド欄に FORT77 JEF LIBDD と入力し，実行キーを押す。
002100 C *****
002200 C
002300 LOGICAL*4 EX
002400 INTEGER*4 RC1,RC2
002500 INTEGER*4 PIA,LSIA(32),RCA
002600 INTEGER*2 MPA,PNA,CSA(2)
002700 CHARACTER*10 GNO(100)
002800 CHARACTER*208 MDA,MDA1
002900 INTEGER*2 LSDA(13)
003000 CHARACTER*80 MIA
003100 CHARACTER*80 CMD1,CMD2,CMD5
003200 CHARACTER*80 MA
003300 CHARACTER*11 CHI/'0123456789'/
003400 CHARACTER*11 CHIN/' 0123456789'/
003500 CHARACTER*27 CHIC/' ABCDEFGHIJKLMNOPQRSTUVWXYZ'/
003600 NCHARACTER*30 MS1,MS2
003700 C ****GKNO=学籍番号;SEI=姓;MEI=名;CKN(8)=中間試験結果;
003800 C ****KMT=期末試験結果;
003900 C ****WT1=中間試験のウェイト;WT2=期末試験のウェイト
004000 C ****HKN=平均点;JNI=順位
004100 CHARACTER*10 GKNO
004200 CHARACTER*20 SEI,MEI
004300 REAL HKN
004400 INTEGER CKN(8),KMT,JNI,WT1,WT2
004500 DATA CMD1/'ALLOC F(FT60F001) DA('AB9999.WORK1.OUTLIST') SH'/
004600 DATA CMD2/'ALLOC F(FT70F001) DA('AB9999.WORK1.OUTLIST') SH'/
004700 DATA CMD5/'ALLOC F(MENULIB ) DA('AB999.XXXX.DATA') SH'/
004800 DATA LSIA/32*0/
004900 DATA MIA /'MENULIB IPF01 '/
005000 DATA LSDA /24,80,1,24,1,80,0,0,0,0,0,0,0,0/
005100 C ***** 入力用データセットの確保と論理機番への結合 *****
005200 C
005300 C ***** 入力用データセットの有無の確認 *****
005400 INQUIRE(FILE='AB9999.WORK1.OUTLIST',EXIST=EX)
005500 IF (EX) GOTO 201
005600 PRINT *,'入力用データセットが存在しません。'
005700 GOTO 9999
005800 C ***** 入力用データセットの確保 *****
005900 201 CALL IPFCMD(RC1,RC2,CMD1,80)
006000 IF (RC1.NE.0) THEN
006100 PRINT *,'*** ALLOC ERROR ***',
006200 & 'RC1=',RC1,' RC2=',RC2
006300 GOTO 9999
006400 ENDIF
006500 C ***** 入力用データセットのレコード数の計算 *****
006600 202 OPEN (60)
006700 MDA=' '
006800 INO=0
006900 401 READ(60,'(A88)',END=402) MDA(1:88)
007000 INO=INO+1
007100 GNO(INO)=MDA(1:10)
007200 GOTO 401
007300 402 CLOSE (60)
007400 CALL IPFCMD(RC1,RC2,CMD2,80)

```

```

007500     IF (RC1.NE.0) THEN
007600         PRINT *, '*** ALLOC ERROR ***',
007700         & 'RC1=', RC1, ' RC2=', RC2
007800         GOTO 9999
007900     ENDIF
008000     OPEN (70, ACCESS='DIRECT', RECL=88)
008100     CALL IPFCMD(RC1, RC2, CMD5, 80)
008200     IF (RC1.NE.0) THEN
008300         PRINT *, '*** ATT FREE ERROR ***',
008400         & 'RC1=', RC1, ' RC2=', RC2
008500         GOTO 9999
008600     ENDIF
008700 C ** DJOH02 の画面のオープン *****
008800     PIA=0
008900     CALL IPFOVS(PIA, LSIA, LSDA, RCA)
009000     IF (RCA.NE.0) THEN
009100         PRINT *, '*** IPFOVS ERROR ***', 'RCA=', RCA
009200         GOTO 9999
009300     ENDIF
009400 C *****
009500 C
009600 C
009700 1000 MS1=NC'検索する学籍番号を入力して下さい。'
009800     MS2=NC'終了はPF 3キーまたはPF 15キーを押して下さい。'
009900     MDA=' '
010000     CSA(1)=1
010100     CSA(2)=1
010200 420 MDA(89:148)=MS1
010300     MDA(149:208)=MS2
010400     CALL IPFMIO(LSIA, MIA, MDA, CSA, RCA)
010500     IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 3000
010600     IF (RCA.NE.0) THEN
010700         PRINT *, '*** IPFMIO ERROR ***', 'RCA=', RCA
010800         GOTO 3000
010900     ENDIF
011000     DO 403 IO=1, INO
011100         IF (MDA(1:10).EQ.GNO(IO)) GOTO 410
011200 403 CONTINUE
011300     MS2=NC'該当する学籍番号は存在しません。'
011400     GOTO 420
011500 410 MDA=' '
011600     READ(70, REC=IO) MDA(1:88)
011700     MS1=NC'追加・修正をすることができます。'
011800     CSA(1)=2
011900     CSA(2)=1
012000     MS2=NC' '
012100     MDA(89:148)=MS1
012200     MDA(149:208)=MS2
012300     CALL IPFMRR(LSIA, MA, MPA, MDA, PNA, CSA, RCA)
012400     IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 80
012500     IF (RCA.NE.0) THEN
012600         PRINT *, '*** IPFMRR ERROR ***', 'RCA=', RCA
012700         GOTO 3000
012800     ENDIF
.....

```

プログラムのほとんどの部分が問題－１のプログラムと重複するので、重複する部分は説明を省略します。

2300－ 5000：各変数の宣言と初期値の設定

5400－ 5700：入出力用データセットの有無の確認

5900－ 6400：入力データセットの論理機番 60 への結合

6600－ 7300：入力用データセットのレコード数の計算。INO にレコード数が入ります。また、配列 GNO に学籍番号が入ります。

7400－ 7900：入出力データセットの論理機番 70 への結合。

OPEN (70, ACCESS='DIRECT', REC=88)

は、データセットを論理機番 70 に結合し、アクセスがダイレクトモードであること、すなわち、第 1 レコードから順番に読書きするのではなく、指定したレコード番号のレコードからデータを読んだり、レコードに書き込んだりすることを指定しています。REC=88 はそのレコード長が 88 バイトであることを宣言しています。

8100－ 8600：メニュー定義体のデータセットをファイル名 MENULIB に結合します。

8800－ 9300：メニュー画面のオープン

9700－10300：メニュー画面に表示するメッセージの代入、カーソル位置の指定および MDA の初期化を行っています。

10400－10900：メニュー画面を表示し、学籍番号の入力を促します。

11000－11400：画面から入力した学籍番号が、すでに入力してある学籍番号と一致するものがあるかどうかをチェックしています。一致した場合は、一致したレコード番号が IO となります。一致したものがない場合は、その旨メニュー画面に表示し、再入力を促します。

11500－11600：MDA が初期化され、IO 番目のレコードが MDA に読み込まれます。

11700－12800：表示内容の変更・修正を促します。

12900：プログラム（その 1）の行番号 11200 以降と同様です。

### ★★★★ 問 題 ★★★★★

2. プログラム（その 1）および（その 2）を結合し、1 つのプログラムにまとめてください。そして、メニュー画面で

(1) 各学生の成績入力

(2) 指定した学生の成績追加・修正

の項目番号を指定し、指定された項目の処理をするようにプログラムを追加・修正して下さい。

## 6. その他の I P F サブルーチン

I P Fを利用するときは、それぞれの機能に対応した I P F サブルーチンを C A L L を使用して呼び出します。これまでの例では、最もよく使用される I P F C M D, I P F C V S, I P F M I O, I P F M R R, I P F O V S の 5 種を用いて説明したが、その他に 10 種のサブルーチンが用意されています。これらをまとめて表 3 に示します。

表 3 I P F サブルーチンの呼出し名

呼 出 し 名	機 能 分 類	機 能 の 簡 単 な 説 明
I P F C M D	T S S コマンド処理	T S S コマンドを使用する。
I P F C V S	フルスクリーン型 入出力処理	論理画面クローズ
I P F D T A		データ画面の表示 (オペレータ介入あり)
I P F D T O		データ画面の表示 (オペレータ介入なし)
I P F F I O		フィールド定義体を用いた論理画面のアクセス
I P F F T O		フィールド定義体を用いた論理画面の出力
I P F M E R		メニューエラーの処理
I P F M I O		メニュー定義体を用いた論理画面のアクセス
I P F M T O		メニュー定義体を用いた論理画面の出力
I P F M R R		日本語メニューエラーの処理
I P F O P N		論理画面のオープン (同期モード)
I P F O V S	フルスクリーン型	論理画面のオープン (非同期モード)
I P F P F F	入出力処理	標準スクロールキーの解放
I P F P F S		標準スクロールの再設定
I P F R I O		論理画面のリフレッシュ

これらのサブルーチンの使用の詳細は「I P F 使用手引書」を参照して下さい。

## 7. メニュー定義体のエラーチェック

メニュー定義体にエラーがある場合、プログラムで引用されるまでエラーを発見するのは困難です。また、エラーが指摘された場合でも、その原因を知るには相当の経験が必要です。特に、メニュー原型部の入力域、アクション部に日本語の空白が存在する場合、最初はその発見は困難です。このようなメニュー定義体のエラーをチェックし、また、アクション部のパラメータ番号の誤りの修正を自動的に行う問題解決支援システムが準備されています。まず、READY状態（2ページ参照）でKPFDと入力して下さい。＜金沢大学問題解決支援・拡張機能選択初期画面＞が表れます（図6）。

```

-----< 金沢大学問題解決支援・拡張機能選択初期画面 >----- PFDMAIN -----
オプション ==>
-----
0  ATTRIBUTES - PFD属性及び端末特性を定義する。
1  BROWSE     - データセットの内容を表示／検索する。
2  EDIT       - データセットの内容を編集する。
3  UTILITY    - 各種ユティリティを実行する。
6  TSS        - TSSコマンド及びコマンドプロシジャを実行する。
H  HELP       - PFDの使用方法を表示する。
-----
ユーザID   - JM0365
時刻       - 16:19
端末       - F6650
PFキー     - 24
第一修飾子 - JM0365
-----< 金沢大学拡張機能 >-----
S  SERVICE    - 編集とコマンド実行を主とした多目的機能を支援する。
U  UTILITY    - データセットの操作を主とした多目的機能を支援する。
F  FORTRAN    - FORTRANの実行をTSS又はバッチで行う。
B  BATCH      - バッチジョブ用ジョブ制御文の作成／起動を行う。
C  COMMAND    - TSSコマンドの作成／実行を行う。
M  MANUAL     - 利用の手引の紹介／印刷／例題の複写を行う。
P  PRINT      - データセットの内容を印刷する。
K  KANAZAWA   - センターの各種サービス機能を実行する。
E  ERROR      - エラー・メッセージの表示／検索する。
G  GRAMMAR    - 文法規則の表示／検索する。
D  DOGRACE    - 金沢大学の図形・文法処理システムを起動する。
-----

```

図6 KPFD初期画面

オプション欄にCを入力して実行キーを押して下さい。図6が表れますので、4を番号選択欄に入力して下さい。すると次に、図8が表れます。ここで7を番号選択欄に入力して下さい。図9が表れます。

図8の入力データセット名にチェックしようとするメニュー定義体が入ったデータセット名（メンバー名）を入力し、パラメータ番号の修正、構文の確認、テスト表示の有無を入力してコマンド欄にGO、またはRUNを入力して実行キーを押すと、メニュー定義体のチェックが開始されます。チェックが終了し、実行キーを押すと図10のようにメニュー画面のテスト表示が行われ、入力欄などが明らかとなります。

終了する場合は、終了キー（PF 3キー）を順次押して行くと初期画面に戻ります。

-----< 金沢大学・コマンド・オプションメニュー >-----  
番号選択 ===>

---

1 COMP      高級言語（FORTRAN, PL/I）をTSS下での実行する。  
2 APPL      - ADJUST, ATFのコマンドを実行する。  
3 SAS        SASを実行する。（KSAS COMMAND）  
4 KANAZAWA センターユティリティ（MIOCHK,ASCCNVT,FEDIT）

---

画面を終了する場合は、ENDキーを押して下さい。

図7      コマンドオプションメニュー画面（その1）

-----< 金沢大学・コマンド・オプションメニュー >-----  
番号選択 ===>

---

1 MIOCHK    IPFのメニュー定義体のリナンバーおよび文法チェックを行う。  
2 ASCCNVT   ASCIIコードをEBCDICコードに変換する。  
3 FEDIT     レコード長が255バイトを超える固定形式のデータセットを編集する。

---

画面を終了する場合は、ENDキーを押して下さい。

図8      コマンドオプションメニュー画面（その2）

## おわりに

ここでは画面型入出力ソフトウェア（IPF）の機能およびその使用方法について例題を用いて説明を行いました。まず、プログラムがどのようにして組まれているか、それがどのように引用されているかを理解し、実際にプログラムを走らせていただければ、自方のプログラムへの応用方法が明らかになるものと思います。どうか、このように便利な機能を十分に利用されますことを願っています。

