

会 話 型 デ ー タ 処 理 — そ の 3 — REDIM 文 を 使 っ た 配 列 操 作

吉 井 守 正 (鉱 床 部)
Morimasa YOSHII

1. 配列を可変にする

配列をもつ一団のデータを電子計算機のメモリーに収めるには プログラムの中で配列宣言をして その配列要素に入れるのが普通である。

ところで データの種類によってその配列規模は一般に異なるので 種々のデータをひとつのプログラムで共同的に扱うには 配列規模を大き目に取ればよいだろう。しかし 多量のデータを処理するには 1個でも多くのデータを収容できるよう 工夫する必要が出てくる。

そこで考えたのが 配列宣言によって設定された配列規模 (行数×列数) の範囲内で できるだけ多くのデータを詰め込む方法である。筆者がこれまで話題にしてきた YHP-9845T (横河ヒューレット・パッカード社製 会話型電子計算機) では 拡張 BASIC 言語の中に一度宣言された配列を変更する REDIM 文がある。これを活用すると 上に述べた目的は達成される。

筆者が最近 9845T のために作った 主として岩石研究用の化学分析データ処理システムでは 大ざっぱに言うと 1組の化学分析値 (つまりデータ1個分) が配列の行要素に そして各化学成分が同じく列要素に それぞれ対応している。データの種類 つまり研究目的や対象により 取扱われる化学成分数 (列数) が異なるので 成分数をもとに データの許容数 (行数) を算出して 配列の行数と列数を修正する。このようにすると割り当てられた配列の中にデータを最大限入力できるので メモリーを有効に使えることになる。

では その具体的な方法について つぎに説明しよう。

2. 岩石用化学分析データ入力プログラムでの例

前の回 (本誌317号) で紹介した化学分析データ入力プログラムでの配列規模の変更操作を例にとって説明しよう。

このプログラムで関係する配列名は 化学成分名用に Comp\$ (*) データ番号・コード用に No\$ (*) 化学分析値用に D(*)がある。ここで(*)は配列を意味することにしよう。

プログラムの冒頭での配列宣言は 標準的にはつぎのとおりである (ノルム計算用は配列規模が少し小さい)。

COM Comp\$ (30) [6]……各6文字30成分の文字列配列。

COM No\$ (2400, 2) [10]……各10文字データ数2,400個分の文字列配列 (データ番号・コード用で2列)。

COM SHORT D (1800, 15)……1/2精度による1,800個 15成分の配列 (精度的にはこれで十分)。

これらの配列規模の変更操作のためには その行数や列数を指示するための配列を用意する。これを Sc(*) と呼ぶことにする。

その配列要素は つぎのように定義される。

Sc(1) データの現在数

Sc(2) データの許容数 (入力可能な最大数)

Sc(3) “Total” の項までの化学成分数 (詳しくは本誌317号)

Sc(4) 全化学成分数 (“Total” の項も含む)

この配列 Sc(*) が いわば今回のポイントである。化学分析データ入力プログラムを実行すると データを新規に入力する場合は化学成分名の入力行程がある。成分名は Comp\$ (*) に入れられるが 入力が終わった段階で化学成分数が確定し その数が Sc(4) に記憶される。これをもとに Comp\$ (*) の配列規模が つぎのように変更される。

REDIM Comp\$ (Sc(4))

さらに Sc(4) をもとに データの許容数 Sc(2) が 算出される。プログラムステップとしては

Sc(2)=INT (Rowcom * Colcom/Sc(4)) .

ここで Rowcom と Colcom は 配列宣言したときの D(*) の行数と列数で 上の例ではそれぞれ1,800と15である。ただし Sc(2)の値は No\$ (*) の行数を超える事はできないので 最大2,400である。

Sc(2) と Sc(4) の値をもとに D(*) の配列規模が つぎのように変更される。

REDIM D (Sc(2)
Sc(4)).

データの現在数を示す Sc(1)は 最初は当然0である。キーボードからデータが1組入力されるごとに Sc(1)は1つずつ足し上げられる。これらの入力が終了してデータが磁気テープのファイルへレコードされる際 その直前に No\$ (*)と D (*)の配列規模が Sc(1) を使ってつぎのように変更される。

REDIM No\$ (Sc
(1), 2)

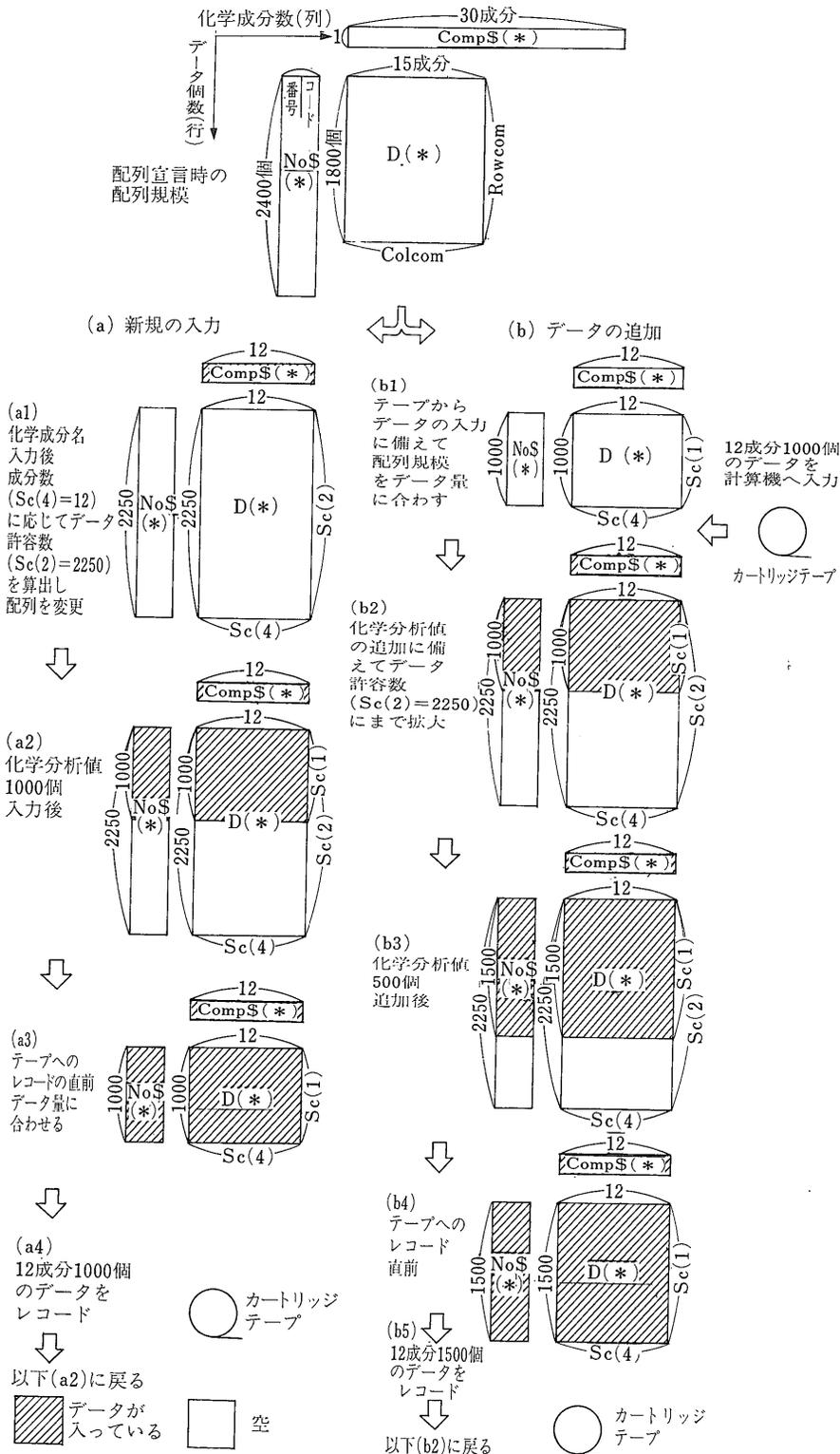
REDIM D (Sc(1),
Sc(4)).

これでこれらの配列規模を データの量と一致させる事ができる。レコードのあと配列規模を再びつぎのように拡大しておく。

REDIM No\$
(Sc(2), 2)

REDIM D (Sc(2),
Sc(4))

これはレコード後にもデータの追加を可能にするためプログラムのステップはこのあとデータの入力・訂正の行程へもどる。多量の



第1図 岩石用化学分析データ入力プログラムでの配列規模変更例
ここでは (a) 新規入力の際12成分のデータを1,000個入力したのちテープにレコードし (b) テープからそのデータを再び計算機に入力して さらに500個追加した後テープへレコードする という想定で図が描かれている。

```

1660 Rcd:      ! Record data
1670 ASSIGN #1 TO Idxf$
1680 ASSIGN #2 TO Idx$(1)
1690 PRINT "INDEX FILE= ";Idxf$;" DATA FILE= ";Idx$(1)
1700 PRINT "Recording is on -- Wait a while."
1710 PRINT #1;Idxf$,Idx$(*),Sc(*)
1720 REDIM No$(Sc(1),2)
1730 REDIM D$(Sc(1),Sc(4))
1740 PRINT #2;Comp$(*),No$(*),D$(*),END
1750 ASSIGN #1 TO *
1760 ASSIGN #2 TO *
1770 PRINTER IS 16
1780 PRINT PAGE
1790 REWIND
1800 REDIM No$(Sc(2),2)
1810 REDIM D$(Sc(2),Sc(4))
1820 GOTO Er
    
```

第2図
データのレコード行程のプログラム
リスト
PRINT #1 PRINT #2 がそれぞ
れインデックスファイルとデータフ
ァイルへのレコード命令。

データを入力するときは 作業をときどき中断して入力されたデータのレコードを繰り返す方が 停電その他の事故でせっかく入力されたデータが消失する危険が少ない。 以上の実例を第1図(a)に模式化して示す。 ここでは成分数12 データ数1,000個の入力を想定する。 レコード行程のプログラムリストを第2図に示す。

3. 磁気テープのファイル構成

計算機に入力されたデータは 一般に最終的には磁気テープのファイルなどに収められる。 配列規模の問題とも密接に関係するので 話を一度中断してテープのファイルについて述べておこう。

筆者の方式では ファイルには2種類あり それぞれ “インデックスファイル” と “データファイル” と呼ぶ。 インデックスファイルには

- 1) そのファイル名 Idxf\$
- 2) 日付・作業名その他の申し送り事項 Idxf(*)
- 3) データの配列規模 Sc(*)

を入れる。 データファイルには

- 1) 化学成分名 Comp\$(*)
- 2) 番号とコード No\$(*)
- 3) 化学分析値 D(*)

をそれぞれ収める。 ファイルの構成を第3図に示す。 使用者が定めるファイル名は インデックスファイル

名だけであり データファイルには インデックスファイル名に序数を付けたものが 自動的に命名される。 たとえば インデックスファイル名が “ALPS” なら データファイルは “ALPS 1”, “ALPS 2”, ……と付番される。 ただし 現在9845Tで使用されているカートリッジテープの容量は約200キロバイトで 筆者のシステムでのデータの最大量は170キロバイト程度だから 目下のシステムでは データファイルは1個分しか作られない。 しかし将来大容量の外部メモリーが計算機に接続される場合に備えて 複数のデータファイルを作れる道が開いてある。

このようにファイルを2種類作って これをペアにして使うのは データをテープから計算機に入力する前に 計算機内にある受け入れ側の配列を データの配列と一致させるためである。 両者の配列規模が一致しないと エラーになる。 この入力行程の際の配列操作について つぎに説明しよう。

4. テープからデータを入力する際の REDIM

データをテープのファイルから計算機に入力する操作はデータ処理プログラムでは必ず行われるし データの入力プログラムでも 既存のデータに新しいデータを追加したり データを訂正・印刷するときに行なわれる。

これらのプログラムでは 前にも述べたとおり 標準的にはつぎの配列宣言が行なわれる。

```

COM Comp $ (30) [6]
COM No $ (2400, 2) [10]
    
```

| | | | | | | | |
|-----------------|--------------------|------------------------|---------------|----------------------|--------------------|-------------------|---------------|
| ファイル名 "ALPS" | ファイル名(照合用) Idxf | データに関する記事 Idx\$(10) | 配列指示 Sc(5) | ファイル名 "ALPS" 1 | 化学成分名 Comp\$(*) | 番号・コード No\$(*) | 化学分析値 D(*) |
| インデックスファイル | | | | データファイル | | | |

第3図 テープファイルの構成
()の中の数字は配列の列数 *は配列規模が不定。

```

500 Load: INPUT "FILE. (#5) (Within 5 characters in this system!"; Idxf#
510 PRINT "INDEX FILE= "; Idxf#
520 Idx#(1)=Idxf#&&VAL$(Filen)
530 IF Chi=1 THEN 670
540 ASSIGN #1 TO Idxf#
550 READ #1; Idxf#, Idx#(*), Sc(*)
560 ASSIGN #2 TO Idx#(1)
570 PRINT "DATA FILE= "; Idx#(1)
580 REDIM Comp$(Sc(4))
590 REDIM No$(Sc(1), 2)
600 REDIM D$(Sc(1), Sc(4))
610 READ #2; Comp#(*), No#(*), D#(*)
620 REWIND
630 REDIM Comp$(Sc(4))
640 REDIM No$(Sc(2), 2)
650 REDIM D$(Sc(2), Sc(4))

```

第4図 岩石用化学分析データ入力プログラムでのテープからの入力行程

READ #1 READ #2 がそれぞれインデックスファイルとデータファイルからの読み取り命令。入力プログラムではとくに REWIND (テープの巻戻し) のあとに配列規模拡大のための REDIM 文が続く。

COM SHORT D (1800, 15)

テープからは 最初にインデックスファイルの内容が入力される。これには Sc(*) が含まれているのではじめに宣言された配列は それを使ってつぎのように改訂される。

```

REDIM Comp$( Sc(4) )
REDIM No$( Sc(1), 2 )
REDIM D( Sc(1), Sc(4) )

```

これで データファイル内のデータの配列規模と一致させることができたので つぎにデータファイルの内容が Comp\$(*) No\$(*) D(*)に入力される(第4図)。

データ処理のプログラムでは 配列規模はこのあととは変更されない。データ入力プログラムでは データの追加に備えて No\$(*)と D(*)の行数が データ許容数まで拡大される。すなわち Sc(2)を使って

```

REDIM No$( Sc(2), 2 )
REDIM D( Sc(2), Sc(4) )

```

そして化学分析値の入力行程へと続く。

以上の配列操作の例を第1図(b)に示す。この例ではレコードされた1,000個のデータを再び計算機に入力してこれに500個追加する。これら計1,500個のデータをテープにレコードするという想定である。

5. まとめ

データの量に即応してメモリーの配列規模を動的に変化させるというやり方は 今回急に思いついたのでは

ない。9845Tの前の機種である9820Aには配列機能がなかったので 筆者が配列要素とその計算機のレジスタ番号を対応させる“配列用サブルーチン”を作って配列をもつ多量なデータの処理を可能にした(本誌275号)。9820Aが以来今日までデータ処理機として盛んに使われてきたゆえんでもある。

さて配列用サブルーチンで配列の割り付けをする時たとえば配列の列数を固定すると行数については自由に増減できることになる。この性質に目を付けて配列の行要素を各データに対応させるとデータ数の増減と合わせて配列の行数を可変できる(本誌275号)。これがメモリーの有効利用につながるのである。今回はこの考えを9845T用にも採用したというわけである。

さらにファイルを上に述べた2種類にしてインデックスファイルの入力によって配列規模を変更させる方法も9820A用のプログラムシステムでは常用している。このように以前に採用して成功をおさめた手法は新しいシステムでも踏襲してそれに新しい考えなり機能なりを付け加えるという手堅い方法を筆者としては行なっている。

これまで3回に分けてご紹介して来た内容は筆者流システムの基本とでも言うべき部分でこれを自動車に例えればエンジンとシャーシーだけである。あとは目的に合った車体(サブルーチン)を乗せればよいのである。およそ配列をもったデータなら基本的なステップを少し変更し入出力のフォーマットを変えるだけでかなり応用できるよう工夫したつもりである。

これらの応用例として岩石のノルム計算を行なって結果をさまざまに処理するプログラムも作られ順次改良されている。