1　ポインタと二次元配列

学習のポイント

ポインタを用いて二次元配列を参照する方法を学びます。

二次元配列はメモリ上には、下図の物理イメージのように一次元的に、連続して格納されています。つまり a[0][0] を先頭に、a[0][1]、・・・a[0][4]、a[1][0]・・・というように列要素が先に変化する順序で格納されています。

さて、ポインタを用いて二次元配列を参照するには、

```
int *pa;
pa = a[0];
```

とします。a[0] は二次元配列の先頭のアドレスを示すポインタ定数と考えられ、この値を pa に代入しているので、*pa とすれば、a[0][0] の内容が取り出され、以後 pa の値を進めることにより、連続的に配列要素を参照できるわけです。

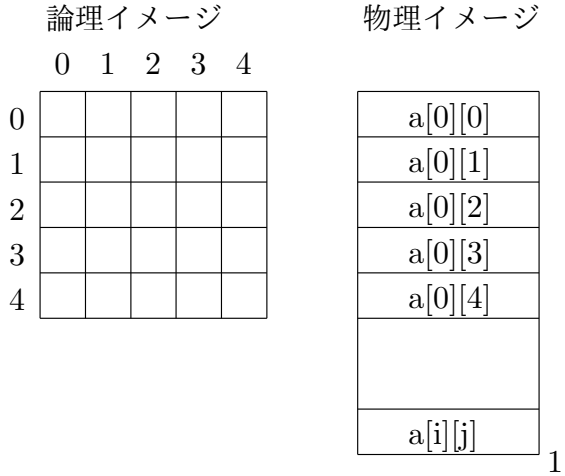
さて、C における二次元配列 a[i][j] の、a、a[i]、a[i][j] はそれぞれ意味を持っています。配列名 a は二次元配列の先頭番地を示す定数であり、a[i] は第 i 行 (a[i][0]、a[i][1]・・・) の先頭番地を示す定数であり、a[i][j] は i 行 j 列の要素を示す変数です。したがって

```
pa = a[1];
```

としておいて、*pa とすれば、これは a[1][0] を参照することになります。

このように、C では配列の参照を柔軟に行えるように a(i, j) という表現を避け、a[i][j] という分離可能な表現をとっているのです。なお、a や a[i] はアドレスを示す定数ですから、これへの代入は行うことができませんので注意してください。

「C 言語」(河西朝雄著　ナツメ社)90 頁



1.1 例題 18

二次元配列 a[ ][ ] のデータをポインタを用いて取り出しなさい。
配列データの終わりは-999 とする。

```
/*
  例題 18 C 言語 91 頁
  二次元配列 a[][] のデータをポインタを用いて取り出しなさい。
  配列データの終わりは-999 とする。
  reidai18.c
*/

#include <stdio.h>

int main()
{
  static int a[][4] = {{ 1, 2, 3, 0},
                       { 4, 5, 6, 0},
                       { 7, 8, 9, -999}};
  int *pa;

  pa = a[0];
  while(*pa != -999){
    printf("%d ", *pa);
    pa++;
  }

  return 0;
}
```

1.2 練習問題 18

二次元配列 a[ ][ ] の第 1 行の要素以後を 0 にするブロググラムを作りなさい。

```
/*
  練習問題 18 C 言語 92 頁
  二次元配列 a[][] の第 1 行の要素以後を 0 にするブロググラムを作りなさい。
  rensyu18.c
*/

#include <stdio.h>

int main()
{
  static int a[][4] = {{ 1, 2, 3, 0},
                       { 4, 5, 6, 0},
                       { 7, 8, 9, -999}};
  int j, k, *pa;

  pa = a[1];
  while(*pa != -999){
    *pa = 0;
    pa++;
  }

  for(j = 0; j < 3; j++){
    for(k = 0; k < 4; k++){
      printf("%5d", a[j][k]);
    }
    printf("\n");
  }
  return 0;
}
```