

# 1 構造体の一括代入 C 言語 162 頁

## 1.1 学習のポイント

構造体データを一括して別な変数に代入する方法を学びます。

struct eisei 型の変数 a, b が、

```
struct eisei a, b;
```

と宣言されており、変数 a に次のようなデータが格納されているものとします。

```
a.name = 'I';  
a.kodo = 5;  
a.shuki = 1.7691;
```

変数 a の各メンバのデータを変数 b にコピーするには、

```
b.name = a.name;  
b.kodo = a.kodo;  
b.shuki = a.shuki;
```

のように各メンバ単位に代入を行なってもかまいませんが、

```
b = a;
```

のように全メンバを一括して代入することもできます。

「C 言語」(河西朝雄著 ナツメ社)162 頁

## 1.2 例題 38 reidai38.c

練習問題 38 木星の 4 大衛星のデータをキーにして大きい順に並べ変えて表示するプログラムを作りなさい。

```
/*  
 例題 38  
 例題 37 の構造体配列 a[] のデータを後のレコードから b[] にコピーしなさい。  
  
  reidai38.c  
*/
```

```

#include <stdio.h>

int main()
{
    int i;

    struct eisei{
        char name;
        int kodo;
        double shuki;
    };

    static struct eisei a[] = {{'I', 5, 1.7691},
                                {'E', 6, 3.5512},
                                {'G', 5, 7.1545},
                                {'C', 6, 16.6890}};

    struct eisei b[4];

    for(i = 0; i < 4; i++){
        b[i] = a[3 - i];
    }

    for(i = 0; i < 4; i++){
        printf("%c%3d%8.4f\n", b[i].name, b[i].kodo, b[i].shuki);
    }

    return 0;
}

```

## 2 AI による大規模データ処理入門

### 2.1 遺伝アルゴリズムの実際

簡単な例題を用いて、遺伝アルゴリズムの具体的な流れを説明します。今、次のような問題を考えます。

問題:数当てパズル

秘密にされた 8 桁の 2 進数がある。この 2 進数を推測し、秘密の 2 進数を求めよ。ただし、推測は何回でも行なえる。また推測結果に対して、何桁合っているかについてのヒントをその都度得ることができるものとする。

上記問題で、秘密の 2 進数は何でもかまいませんが、たとえばここでは、

秘密の 2 進数 (正解の値) 10101010

としておきましょう。

このパズルを解くには、推測値として適当な 8 桁の 2 進数を生成し、何桁合っているのかのヒントを頼りに、解をよりよくしていく必要があります。そこで、8 桁の 2 進数を遺伝子の表現として用いることにします。

AI による大規模データ処理入門 167 頁 小高知宏著

問題:以下の遺伝子の初期集団 (遺伝子プール) を定義して、表示するプログラム (initpool.c) を作成しましょう。

遺伝子プールは、

```
0,0,1,1,0,1,1,0
1,0,1,1,0,0,1,1
0,1,1,1,0,1,1,0
1,0,0,1,0,1,1,1
```

遺伝子配列は、

```
int pool[4][8]
```

として下さい。

## 2.2 遺伝子プールの作成 initpool.c

```
/******  
/*遺伝子プールの初期化          */  
/*initpool.c                    */  
/******  
  
#include <stdio.h>  
  
int main()  
{  
    /*遺伝子プール*/
```

```
int pool[4][8] = {{0,0,1,1,0,1,1,0},
                  {1,0,1,1,0,0,1,1},
                  {0,1,1,1,0,1,1,0},
                  {1,0,0,1,0,1,1,1}};

int i;
int j;

/* プールの状態表示 */
for(i=0;i<4;++i){
    for(j=0;j<8;++j){
        printf("%d ",pool[i][j]);
    }
    printf("\n");
}

return 0;
}
```