

配列

計算機とアルゴリズム I

第 13 回

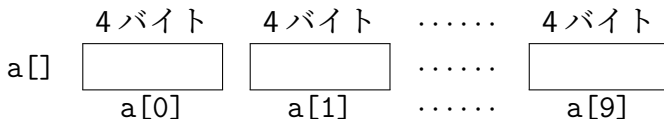
配列

同種のデータ型を連続してメモリ上に確保したものを**配列**という。
(文字列と同様で、宣言の仕方も同様である。)

例 `int a[10];`
 `double b[10];`

比較 `char c[10];`

変数名の後の[]の中の数はサイズであり、サイズ10で宣言した場合、添字は0から9となる。



このとき、各箱には `int` 型の数が1つずつ格納される。

配列

例えば

```
a[0]=1;a[1]=1;a[2]=a[1]+a[0];a[3]=a[2]+a[1];
```

のように、 $a[0], \dots, a[3]$ に値を代入すると、

a[]	1	1	2	3		?
	a[0]	a[1]	a[2]	a[3]		a[9]

となっている。

この例は、フィボナッチ数列の第3項までを表している。

$$a_0 = 1, a_1 = 1, a_{n+2} = a_{n+1} + a_n$$

そこで、もう少し先まで求めるプログラムを考えてみてほしい。

課題 L.1.

自然数を入力し (n とする)、フィボナッチ数列の第 n 項を表示するプログラムを作成せよ。

入力する数を大きくすると正しく表示されない。ことも確認せよ。

課題 L.1.

```
#include<stdio.h>
int main(void){
    int i,n,a[100];
    a[0]=1;a[1]=1;
    printf("自然数を入力してください:");
    scanf("%d",&n);
    .....
    .....
    .....
    return 0;
}
```

配列へ入力データの格納

標準入力 (キーボード) から入力されたデータを配列へ格納することはよく用いられる。

例

```
#include<stdio.h>
int main(void){
    int i, data[100];
    for(i=0; i<10; i++)
        scanf(" %d",&data[i]);
    for(i=0; i<10; i++)
        printf("%d,",data[i]);
    return 0;
}
```

配列

♠ 補足 配列は1次元のものだけでなく、2次元、3次元も扱える。

例えば、2次元配列

```
int a[3][4];
```

は数学の行列ということも出来る。この場合も、配列の添え字に注意すること。

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

配列

例 (2つのデータをまとめて扱う例). 2つのデータをまとめて扱う場合に配列を用いる。(実際は構造体を利用?)

例えば、 n 人の身長と体重をデータとして扱う場合、 $2 \times n$ の配列

$$\begin{array}{cccccc} a[0][0] & a[0][1] & a[0][2] & \cdots & a[0][n-1] \\ a[1][0] & a[1][1] & a[1][2] & \cdots & a[1][n-1] \end{array}$$

を用意する。 i 番目の人の身長を $a[0][i]$ に、体重を $a[1][i]$ に格納しておく。

番号	0	1	2		$n-1$
身長	165.3	173.8	152.5		163.0
体重	58.7	72.1	53.2		62.3

配列

このデータを身長の高い人順に並べ替えてみる。

$i < j$ に対して、 $a[0][i] > a[0][j]$

なら、

- ・ $a[0][i]$ の値と $a[0][j]$ の値と入れ替え
- ・ $a[1][i]$ の値と $a[1][j]$ の値と入れ替え

を行う (ソートについては後述)。

配列 (文字列)

また、文字列の配列は2次元配列のようになる。例えば、

```
char text[4][30];
```

と宣言して、データを scanf 文などで読み込むと

	[0]	[1]	[2]	[3]	[4]	
text[0]	'O'	'k'	'a'	'y'	'a'	
text[1]	'O'	'U'	'S'	0	'?	
text[2]	'1'	'2'	'3'	'4'	'5'	
text[3]	'q'	'w'	'e'	'r'	't'	

のようになる。

課題 L.2.

5 人の名前をアルファベットで入力し、入力後に順に表示させるプログラムを入力、実行せよ。

課題 L.2.

```
#include<stdio.h>
int main(){
    int i;
    char name[5][30];
    for(i=0;i<5;i++)
        scanf(" %s",name[i]);
    for(i=0;i<5;i++)
        printf("%s¥n",name[i]);
    return 0;
}
```

初期化

変数を宣言するときに、最初の値を設定することができる。これを**初期化**という。例えば、以下のような初期化を考える。

```
3 int a=20;
4 double b=3.14;
5 char c='A';
6 int d[3]={1,2,3};
7 int e[2][3]={{1,2,3},{4,5,6}};
8 char f[]={'A','B','C'};
9 char g[10]="abcdef";
10 int h[3]={0,1,2,3};
11 char i[3]="abcde";
```

このとき、10, 11 はエラーとなる。

初期化

(再度注意) 初期化の時には文字の代入のような作業が行われているが、変数宣言後に

```
i[10]="abcde";  
i[]="abcde";
```

のようなことは出来ない。

初期化

- ♠ **補足** 初期化をしなかった場合、不定値となる。
不定値のまま変数値を使おうとすると、通常は警告が表示される。
応用数学科計算機室で使用する環境では警告は出ない。
さらに、int 型なら 1 として扱われる。

```
#include<stdio.h>
int main(){
    int i;
    char c;
    printf("%d,%c",i,c);
    return 0;
}
```

初期化

例 (初期化した配列の要素を表示する例)

```
#include<stdio.h>
int main(void){
    int i;
    int data[10]={75,85,63,59,91};
    for (i=0; i<5; i++){
        printf("data[%d] = %d\n", i, data[i]);
    }
    return 0;
}
```

初期化した場合も for 文や scanf 文で配列を作成したときと同様に動作させることができる。

データのソート [例 1.2.3 参照]

5つのデータを初期化で定義

```
int data[10]={75,85,63,59,91};
```

して、値の小さい順に並び替える。

【考え方】 与えられた5個のデータに対して、次のように考える。

- まず `data[0]` と `data[1]` の大小を調べる。
- もし、`data[0] > data[1]` ならば、`data[0]` と `data[1]` を入れ替え、
 `data[0] < data[1]` ならば、次の `data[1]` と `data[2]` の大小を調べる。
- 入れ替えを行ったら最初から大小を調べる。
- `data[3]` と `data[4]` まで比べて入れ替えが無ければ終わる。

データのソート [例 1.2.3 参照]

【アルゴリズム】 少しプログラムの考えにする。

1) まず $i=0$ とする。

2) i が 4 未満の範囲で、 $\text{data}[i]$ と $\text{data}[i+1]$ の大小を調べる。

2-1) $\text{data}[i] > \text{data}[i+1]$ ならば、 $\text{data}[i]$ の値と $\text{data}[i+1]$ の値を入れ替える。

入れ替えたら、最初からまた比べる ($i=0$ として 2) に戻る)。

2-2) $\text{data}[i] \not> \text{data}[i+1]$ ならば、 $i++$ をして 2) に戻る。

3) i が 4 となり、終了。

(i が 4 になったら交換が無かったことを意味する。)

例 5.1.5'.

例 (与えられたデータを大きい順に並べ替える)

```
#include<stdio.h>
int main(void){
    int i,temp,data[10]={75,85,63,5,91};
        (※ 1)
    i=0;
    while(i<4){
        if(data[i]>data[i+1]){
            temp=data[i];
            data[i]=data[i+1];
            data[i+1]=temp;
            i=0;
        }
        else
            i++;
    }
        (※ 1)
    return 0;
}
```

例(与えられたデータを大きい順に並べ替える)

実行結果

```
75,  85,  63,   5,  91,  
 5,  63,  75,  85,  91,
```

(※ 1) 部分は printf 文である。例えば、上記のように表示なら

```
for(i=0;i<5;i++)  
    printf("%4d,",data[i]);  
printf("¥n");
```

のようなプログラムとなる。

本日の課題

- ・課題 L.1.
- ・課題 L.2.
- ・課題 L.3. (プリント 課題 5.3.1.)
- ・課題 L.4. (プリント 課題 5.3.6.)

提出方法はメールで yoshiyuki-mori@ous.ac.jp まで送る。

注意！6月からアドレス変更。

件名は学生番号と課題 L. と書き、

例) S24M999 課題 L.

ファイルを添付し送ること。

締め切りは7月20日23:59

7月19日の講義はVODでも対応します(詳細はホームページで)。