プログラミング演習 第9回 配列II
並び替え (バブルソート)

2000年6月8日

目 次

1 今までの講議に関する確認と補足 **1**

1.1 式の値 1

1.2 配列とポインタ 2

2 最大値捜し **4**

3 並べ替え (バブルソート) **4**

4 演習 **6**

5 練習問題 **7**

1　今までの講議に関する確認と補足

1.1　式の値

以前に代入文は，式自体が値を持つことを学びました．同様に，条件式も値を持ちます．while 文を例に考えてみましょう．

```
while (1) {
    printf("infinite loop!\n");
}
```

この場合，while の中の中括弧で囲まれたブロックは，永遠に繰り返されます．一般に

```
while (式) {
    ....
}
```

では，式の値を計算し，それが 0 でない場合，ブロックの中を実行します．条件式はそれが真のとき，0 でない値を取り (1 を取る場合が多い)，偽のときは，0 を取ります．従って，

```
a = 0;
while (a < 3) {
    a = a + 1;
}
```

というプログラムでは，a が 0, 1, 2 のときの 3 回繰り返したあと，a < 3 の値は，**0** になり，次は，while の中の文は，実行せずに，その下へとプログラムは進みます．これを利用すると，次のようなプログラムを組むことも出来ます．

```
a = -3;
while (a) {
    a = a + 1;
}
```

文字列の最後には必ず数としての 0, C 言語の文字記号で　'\0' が付きます (null terminated string と言う). また, 文字列の初期化ではダブルクォーテーション " で囲む事により, 最後の　'\0' が自動的付加されます. これを利用すると, 次のような場合 while 文が問題無く実行されます. すなわち, \0(null) 以外はループで回し, 終端の \0 が来た時点でループは終了します.

```
#include <stdio.h>

main()
```

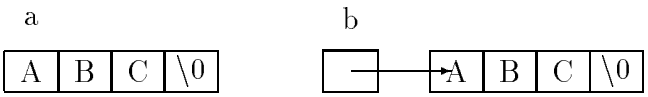
```
{
        char *p="This is a test.";

        while(*p){
                putchar(*p);
                p++;
        }
}
```

1.2 配列とポインタ

配列を宣言すると，配列名はその配列の先頭番地を示すポインタとして扱われました．
例えば,

```
char a[] = "ABC";
char *b = "ABC";
```

の違いは次の図から理解できると思います:



a, b とも配列の先頭を指すポインタでありますが, b はポインタ型変数のため`b +1` とすると意味がありますが (この場合矢印が 1 つ進み B のアドレスを指す), `a+1` は意味がありません. 但し, `*(a+1)` は意味があり, B を指します.

関数に配列名を渡すと，その先頭番地がコピーされるので，関数の中で，配列の要素の値を操作することが出来ます．

次のプログラムは，何を行っているか考えてみましょう．

```
void swap_aij(int *pa, int i, int j);
main ()
{
    int  a[10];
    ....
    swap_aij(a, i, j); /* a は配列の先頭番地 */
    ....
}

void swap_aij(int *pa, int ii, int jj)
{
```

```
    int    tmp;

    tmp = *(pa + i);        /* tmp = pa[i] と書いても同じ */
    *(pa + i) = *(pa + j);  /* pa[i] = pa[j] と書いても同じ */
    *(pa + j) = tmp;        /* pa[j] = tmp と書いても同じ */

    return;
}
```

補足

先週の「内積を用いた角度の計算」のプログラムをポインタで書き直してみますと以下の通りになります.

```
#include <stdio.h>
#include <math.h>

double inn_prod(double *, double *);

main()
{
        char nyuryoku[80];
        double a[3], b[3], *pa, *pb;
        double angle;

        pa = a;        /* ポインタデータの代入*/
        pb = b;       /* ポインタデータの代入*/

        printf("最初のベクトルの x, y, z 成分を空白で区切って入力して下さ
い>>");
        gets(nyuryoku);
        sscanf(nyuryoku, "%lf %lf %lf", &a[0], &a[1], &a[2]);

         printf("次のベクトルの x, y, z 成分を空白で区切って入力して下さい
>>");
        gets(nyuryoku);
        sscanf(nyuryoku, "%lf %lf %lf", &b[0], &b[1], &b[2]);

       angle = acos(inn_prod(pa, pb)/sqrt(inn_prod(pa, pa)*inn_prod(pb, pb)));

        printf("2 つのベクトルのなす角はおよそ %f ラジアンです\n", angle);
```

```
                }

    double inn_prod(double *pa, double *pb)
                    {
            return (*pa)*(*pb) + (*(pa+1))*(*(pb+1))+ (*(pa+2))*(*(pb+2));
          }
```

2 最大値捜し

与えられた n 個の整数から最大数を求めるプログラムを配列 $d[n]$ をつかって組んでみると, 主要部分は以下の通りになります:

```
for(i=0;i < n; i++)
{  if(d[i] >= d[i+1])
      tmp = d[i];
      d[i]= d[i+1];
      d[i+1] = tmp;
}
```

tmp は, for 文の前で定義されている, 整数型の変数.
この方法を用いると, 最大数は最後のd[n-1] であることがわかります.
もちろん, このほかの方法もありますが, あとで利用するため上の例をあげました.
基本的アイデアは, 2 つの整数の順序を取り替えるアルゴリズム

```
tmp = a;
a = b;
b = tmp;
```

で, 予備の変数tmp を用いて取り替えをしなければなりません.(何故か)

3 並べ替え (バブルソート)

与えられた整数を小さい順に並び替えるバブルソートについて今回学びます.
具体的にどういうものかと言いますと, 今,

```
37 23 11 19 5 7
```

という 6 個の整数が与えられたとき, これを小さい順に並び替えて,

```
5 7 11 19 23 37
```

にするということです.
実際, どのようなアルゴリズムをするでしょうか?

(1) まず一番目の 37 を 2 番目の 23 と比較する. $37 > 23$ より, これをとりかえる.

(2) 次に 2 番目の整数(今の場合は 37) と 3 番目の整数を比較し, 3 番目の整数が小さかったら取り替える.

(3) 以下これを繰り返すと,

```
23 11 19 5 7 37
```

となります.

(4) 次に, 上の操作 (1) - (3) を前から 5 つの整数に行う. そうすると,

```
11 19 5 7 23 37
```

となります.

(5) 以下, 順に比較する整数を減らして, (1) - (3) の操作を行う.

つまり, 比較する整数の集合を配列 $d[n]$ であらわしたとき, 上のアルゴリズムは,

```
for(i = n-1; i>= 0; i--)
{d[0] と d[2], ....,d[i] を比較し, 最大数を d[i] とする}
```

となります.
それでは実際に組んでみましょう.

4　演習

以下の問題を時間内に大坂までメールで送ること．但し, Subject 名は creportno.9-1 とする.

与えられた $n(500$ 以下$)$ 個の整数を小さい順に並び替える次のプログラムを完成せよ.

```
#include<stdio.h>
#define MAX 500

main()
{
        int d[MAX];
        int n, tmp, i, j;

        printf("並べ替える整数の個数を入力せよ\n");
        scanf("%d", (1));

        printf("%d 個のデータを入力してください\n",n);

        for((2))                    /* データ入力 */
        { (3)}

        for(i=n-1; i>=0; i--)       /* 並び替え */
         {
            for((4))
            { (5)}
          }

        for(i=0;i < n; i++)         /* データ表示 */
         { (6)}
}
```

注意

時間内に提出できなかった人は, 6 月 20 日までに提出して下さい.

5　練習問題

余裕のある人は次のプログラムを作成せよ.

(1) 先のcreportno.9-1をint d[MAX]を大きい順に並びかえるvoid型関数`void sort_data(void)`を定義して書き換えよ.
(2) 文字列を入力し, その文字列を逆アルファベット順に並べ替えた文字列を出力するプログラムを作成せよ.　例えば, chisato を並べ替えるとtsoihca となる.

Subject名はcreportno.9-2, creportno.9-3とし, 締切は6月20日とします. 提出した人は評価いたします.