

配列は、

```
int seisu [10];
```

の様に定義して使うことができます。C 言語では、整数型の他に float、double、char 等の変数を定義しているので整数型の配列と同様に、

```
char   moji[10];  
int    tokuten[9];  
float  kion[10];  
double uriage[10];
```

と定義できます。なぜ、配列が便利なのか。例えば、ある日のプロ野球試合の得点を表示プログラムを作るとしましょう。

プログラムでは、

```
int yomiuri;  
int hansin;  
int hirosima;
```

と定義するでしょう。配列を使えば

```
int tokuten[3];
```

と定義できて、順番に表示する時は、

```
for(i = 0; i < 3; i++){  
    printf("%d\n", tokuten[i]);  
}
```

と書けるので便利ですが、変数が 3 個程度ではありがたみは有りません。

1 バイト文字は、char 型で定義されています。

```
char moji;
```

C 言語には、文字列型が無いので、以下のように定義して使います。

```
char *string = "Hello";  
  
printf("%s", string);
```

または、

```

char string[6];

string[0] = 'H';
string[1] = 'e';
string[2] = 'l';
string[3] = 'l';
string[4] = 'o';
string[5] = '\0';

for(i = 0; i < 5; i++){
    printf("%c", string[i]);
}

```

で表示することができます。

## 1 ポインタと文字列

### 学習のポイント

ポインタを用いて文字列を操作する方法を学びます。

C の文字列は””を用いて表します。たとえば”Hello”という文字列はメモリ上に、次のように格納されています。

H	e	l	l	o	\0
---	---	---	---	---	----

\0 は文字列の終わりを示す印で、C コンパイラが”Hello”という文字列をメモリ上に格納する際に自動的に付加します。

さて、C で文字列を扱う場合は、次のようにポインタを用いると便利です。

```

char *str;
str = "Hello";

```

str は char へのポインタで、str=”Hello”; により、”Hello”を格納してあるメモリ上の先頭アドレスがポインタ str に代入されます。つまり、”Hello”という文字列の実体が、str に代入されるわけではありません。

今、ポインタ `str` は、文字列”Hello”の先頭を指しているので、`*str` とすると文字’H’を取り出せます。ここで `str++` とポインタを進めると、ポインタ変数 `str` は、次の文字を指すので、この時`*str` とすると次の文字’e’を取り出すことになります。

「C 言語」(河西朝雄著 ナツメ社)84 頁

## 2 例題 15

文字列の文字を 1 文字ずつ取り出して表示しなさい。

```
/*
 例題 15 C 言語 85 頁
 文字列の文字を 1 文字ずつ取り出して表示しなさい。
  reidai15.c
*/

#include <stdio.h>

int main()
{
    char *str;

    str = "Hello";

    while(*str != '\0'){
        printf("%c\n", *str);
        str++;
    }

    return 0;
}
```

「C 言語」(河西朝雄著 ナツメ社)85 頁

### 3 リターンとリスク

今回はコーポレート・ファイナンスで株価のリターンとリスクの計算方法を説明しました。今回は、身近な上場企業の株価を使って、リスクとリターンを求めてみましょう。株価は2012年9月から2014年10月までの月次の株価を利用します。以下が株価のファイルとプログラムです。

```
日付 始値 高値 安値 終値 出来高 古野電気 6814 furuno.txt
2014年10月 1096 1116 1020 1108
2014年9月 1026 1110 950 1083
2014年8月 829 1045 822 1025
2014年7月 690 899 668 839
2014年6月 665 700 656 681
2014年5月 696 731 636 658
2014年4月 713 739 650 690
2014年3月 669 732 617 710
2014年2月 730 738 637 670
2014年1月 741 810 720 743
2013年12月 683 744 661 737
2013年11月 646 684 620 680
2013年10月 651 720 640 641
2013年9月 656 696 613 654
2013年8月 674 740 640 664
2013年7月 761 880 670 684
2013年6月 800 800 625 746
2013年5月 791 898 750 813
2013年4月 600 799 540 771
2013年3月 494 620 473 600
2013年2月 491 522 455 480
2013年1月 415 496 390 495
2012年12月 326 402 324 400
2012年11月 310 337 301 317
2012年10月 308 315 294 310
2012年9月 308 314 293 308
```

```
日付 始値 高値 安値 終値 六甲 バター 2266 rokko.txt
2014年10月 1114 1114 1070 1083
2014年9月 941 1148 941 1106
```

2014年8月 884 940 880 939  
2014年7月 895 898 875 892  
2014年6月 879 895 862 887  
2014年5月 870 888 851 873  
2014年4月 866 890 845 867  
2014年3月 779 873 763 871  
2014年2月 777 800 744 775  
2014年1月 843 853 779 785  
2013年12月 880 885 829 844  
2013年11月 831 884 810 877  
2013年10月 787 898 782 831  
2013年9月 740 793 720 788  
2013年8月 765 810 736 742  
2013年7月 728 820 716 785  
2013年6月 719 733 695 729  
2013年5月 695 733 686 718  
2013年4月 659 699 630 698  
2013年3月 605 659 600 655  
2013年2月 582 607 580 605  
2013年1月 585 589 560 585  
2012年12月 577 639 573 585  
2012年11月 517 579 510 573  
2012年10月 497 518 490 517  
2012年9月 476 499 472 497  
2012年8月 469 479 467 474  
2012年7月 468 473 457 470

古野電気と六甲バターの株価のリスクとリターンを計算

```
/*
  株価のデータを読み込む
  20141005
  readpri1.c
*/

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

int main(int argc, char *argv[])
```

```

{
FILE *fp1;
FILE *fp2;

char buff[128];
char yyyyymm[17];
int hatune;
int takane;
int yasune;
int owarine1[64];
int owarine2[64];

double p1[64];
double p2[64];
double r1[64];
double r2[64];

double sum = 0.0;
double avr1 = 0.0;
double avr2 = 0.0;
double dev = 0.0;

double rsk = 0.0;
double rtn = 0.0;

double cov = 0.0;

int count1;
int count2;
int i = 0;

if(argc < 3){
printf("error!! 使い方 ./readpric [入力ファイル 1] [入力ファイル
2]\n");
exit(1);
}

//-----

```

```

if((fp1 = fopen(argv[1], "r")) != NULL){

    fgets(buff, 127,fp1);
    printf("%s", buff);
    i = 0;
    while(fscanf(fp1, "%s %d %d %d %d", yyyyymm, &hatune,&takane,&yasune,
                &owarine1[i]) != EOF){
        printf("%s %d %d %d %d\n", yyyyymm, hatune,takane,yasune,owarine1[i]);
        i++;
    }
}else{
    printf("ファイル 1 を開くことが出来ません。 \n");
}

count1 = i;
printf("データ件数は、 %d 件です。 \n", i);

fclose(fp1);

if((fp2 = fopen(argv[2], "r")) != NULL){

    fgets(buff, 127,fp2);
    printf("%s", buff);
    i = 0;
    while(fscanf(fp2, "%s %d %d %d %d", yyyyymm, &hatune,&takane,&yasune,
                &owarine2[i]) != EOF){
        printf("%s %d %d %d %d\n", yyyyymm, hatune,takane,yasune,owarine2[i]);
        i++;
    }
}else{
    printf("ファイル 2 を開くことが出来ません。 \n");
}

count2 = i;
printf("データ件数は、 %d 件です。 \n", i);

fclose(fp2);

//-----

```

```

for(i = 0; i < count1; i++){
    p1[count1 - i -1] = owarine1[i];
}

for(i = 1; i < count1; i++){
    r1[i] = (p1[i] - p1[i - 1]) * 100 / p1[i - 1];
    printf("%5.2lf  ", r1[i]);
}

printf("\n\n");
printf("リターン (平均値)\n");
sum = 0.0;
for(i = 1; i < count1; i++){
    sum = sum + r1[i];
}
avr1 = sum / (count1 - 1);
printf("%5.2lf\n", avr1);

printf("リスク (分散)\n");
for(i = 1; i < count1; i++){
    sum = sum + pow((r1[i] - avr1), 2);
}

dev = sum / (count1 - 1);
printf("%5.2lf\n", dev);

printf("リスク (標準偏差)\n");
printf("%5.2lf\n", sqrt(dev));
//-----

for(i = 0; i < count2; i++){
    p2[count2 - i -1] = owarine2[i];
}

for(i = 1; i < count2; i++){
    r2[i] = (p2[i] - p2[i - 1]) * 100 / p2[i - 1];
    printf("%5.2lf  ", r2[i]);
}

```



```

printf("\n\n");
printf("リターン (平均値)\n");
sum = 0.0;
for(i = 1; i < count2; i++){
    sum = sum + r2[i];
}
avr2 = sum / (count2 - 1);
printf("%5.2lf\n", avr2);

printf("リスク (分散)\n");
for(i = 1; i < count2; i++){
    sum = sum + pow((r2[i] - avr2), 2);
}

dev = sum / (count2 - 1);
printf("%5.2lf\n", dev);

printf("リスク (標準偏差)\n");
printf("%5.2lf\n", sqrt(dev));
//-----
printf("%5.2lf\n", cov);

for(i = 1; i < count1; i++){
    cov = cov + (r1[i] - avr1)*(r2[i] - avr2);
}

printf("共分散\n");
printf("%5.2lf\n", cov / (count1 - 1));

return 0;
}

```

## 4 使い方

```
./readpri1 furuno.txt rokko.txt
```

## 5 使い方

古野電気と六甲バターの株価のリターンとリスクは下記の様に計算されました。

社名	リターン	リスク
古野電気(株)	5.86	11.86
六甲バター(株)	3.28	5.62