

FORTRAN (と C) によるプログラミング

5 ファイル入出力

ここではファイルからデータを読みこんだり、ファイルにデータを書き出したりするプログラムを作成してみます。はじめに、テキスト形式で書かれたデータファイルに書かれているデータを読みこんで配列に代入し、標準出力に書き出すプログラムを作ります。FORTRAN の場合、OPEN 文でファイルを開いた後、標準入力の場合と同様に READ 文でデータを読みこみます。以下のプログラムでは、あらかじめ配列 A を宣言しておき、ファイルから読みこんだデータを配列 A に代入しています。

プログラム：

FORTRAN

```
C PARAMETER文で定数IMAXを宣言する。
C 定数IMAXはデータの個数を表す。
  PARAMETER (IMAX=20)

C 配列Aを宣言する。
  REAL A(IMAX)

C データを読みこむ。

C ファイルを開く。
C 機番は10以降の番号を指定する。
C STATUSは読みこみの場合は' OLD' を指定する。
C FORMはテキストファイルの場合は' FORMATTED' を指定する。
  OPEN(10, FILE=' data05. txt', STATUS=' OLD',
+     FORM=' FORMATTED' )

C ここからデータを読みこむためのDOループが始まる。
C データの個数だけ反復する。
  DO 11 I=1, IMAX

C READ文でデータを読みこむ。
C 機番10を指定する。*は書式を指定しないことを示す。
C データファイルの各行の第1要素が年、
C 第2要素が配列Aに読みこむデータである。
  READ(10, *) IY, A(I)

C ここでDOループ11が終了する。
11 CONTINUE
```

C ファイルを閉じる。

```
CLOSE(10)
```

C ここからデータを書き出すためのDOループが始まる。

C データの個数だけ反復する。

```
DO 21 I=1, IMAX
```

C A(I)の値を標準出力に書き出す。

C 機番6を指定する。

C 書式は'(1X, I4, 1X, F5.1)'を指定する。

C '1X'は1文字の空白、'I4'は4ケタの整数、

C 'F5.1'は全体が5ケタで小数点以下が1ケタの実数という意味である。

```
WRITE(6, '(1X, I4, 1X, F5.1)') 1990+I, A(I)
```

C ここでDOループ21が終了する。

```
21 CONTINUE
```

```
STOP
```

```
END
```

(参考) C

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

```
int main(void)
```

```
{
```

```
/* 変数 imax を宣言する。
```

```
   定数 imax はデータの個数を表す。 */
```

```
int imax=20, i, iy;
```

```
/* 配列 a を宣言する。 */
```

```
float a[imax];
```

```
float aa;
```

```
/* ファイルポインタ fp を宣言する。 */
```

```
FILE *fp;
```

```
/* データを読みこむ。 */
```

```

/* ファイルを開く。
   モードは"r"(テキストファイルの読みこみ)を指定する。 */
fp = fopen( "data05.txt", "r" );

/* ここからデータを読みこむための for ループが始まる。
   データの個数だけ反復する。 */
for (i=0; i<=imax-1; i++)
{

/* 関数 fscanf でデータを読みこむ。
   ファイルポインタ fp を指定する。
   書式は"%d %f"(整数、浮動小数点)を指定する。
   データファイルの各行の第1要素が年、
   第2要素が配列 a に読みこむデータである。
   関数から変数の値を返すときは、変数の値そのもの(たとえば iy)ではなく、
   その変数へのポインタ(たとえば&iy)が返ってくることに注意。 */
fscanf( fp, "%d %f", &iy, &aa );
a[i] = aa;

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

/* ここからデータを書き出すための for ループが始まる。
   データの個数だけ反復する。 */
for (i=1; i<=imax; i++)
{

/* a[i]の値を標準出力に書き出す。
   書式は" %4d %5.1f"を指定する。
   "%4d"は4ケタの整数、
   "%5.1f"は全体が5ケタで小数点以下が1ケタの浮動小数点という意味である。 */
printf( " %4d %5.1f\n", 1990+i, a[i-1] );

/* ここで for ループが終了する。 */
}

```

```
return 0;
}
```

実行例：

```
/home/snaoki> f77 prog05_1.f
/home/snaoki> ./a.out
1991  6.3
1992  6.8
1993  6.2
<中略>
2010  7.0
```

次に、読みこんだ値の平均値を計算し、それぞれの値から平均値を差し引いた値を計算するプログラムを作成します。あらかじめ配列AとBを宣言しておき、配列Aにデータファイルから読みこんだ値を代入します。次に、平均値を計算し、配列Aの各要素の値から平均値を差し引いた値を配列Bに代入して、標準出力に書き出します。

プログラム：

FORTRAN

```
C PARAMETER文で定数IMAXを宣言する。
C 定数IMAXはデータの個数を表す。
  PARAMETER (IMAX=20)

C 配列A、Bを宣言する。
  REAL A(IMAX), B(IMAX)

C データを読みこむ。

C ファイルを開く。
C 機番は10以降の番号を指定する。
C STATUSは読みこみの場合は' OLD' を指定する。
C FORMはテキストファイルの場合は' FORMATTED' を指定する。
  OPEN(10, FILE=' data05. txt', STATUS=' OLD',
+     FORM=' FORMATTED' )

C ここからデータを読みこむためのDOループが始まる。
C データの個数だけ反復する。
  DO 11 I=1, IMAX

C READ文でデータを読みこむ。
```

C 機番10を指定する。*は書式を指定しないことを示す。

C データファイルの各行の第1要素が年、

C 第2要素が配列Aに読みこむデータである。

```
READ(10,*) IY, A(I)
```

C ここでDOループ11が終了する。

```
11 CONTINUE
```

C ファイルを閉じる。

```
CLOSE(10)
```

C 変数Sにゼロを代入する。

```
S = 0.
```

C ここからDOループ21が始まる。

```
DO 21 I=1, IMAX
```

C 変数SにA(I)の値を加える。

```
S = S + A(I)
```

C ここでDOループ21が終了する。

```
21 CONTINUE
```

C 平均値を計算する。

C 関数REALで整数型の変数を実数型に変換する。

```
AVE = S / REAL(IMAX)
```

C ここからDOループ31が始まる。

```
DO 31 I=1, IMAX
```

C A(I)と平均値との差を計算する。

```
B(I) = A(I) - AVE
```

C ここでDOループ31が終了する。

```
31 CONTINUE
```

C ここからデータを書き出すためのDOループが始まる。

C データの個数だけ反復する。

```
DO 41 I=1, IMAX
```

- C B(I)の値を標準出力に書き出す。
- C 機番6を指定する。
- C 書式は'(1X, I4, 1X, F5.1)'を指定する。
- C '1X'は1文字の空白、'I4'は4ケタの整数、
- C 'F5.1'は全体が5ケタで小数点以下が1ケタの実数という意味である。

```
WRITE(6, '(1X, I4, 1X, F5.1)') 1990+I, B(I)
```

- C ここでD0ループ41が終了する。

```
41 CONTINUE
```

```
STOP
```

```
END
```

(参考) C

```
#include <stdio.h>

int main(void)
{

/* 変数 imax を宣言する。
   定数 imax はデータの個数を表す。 */
int imax=20, i, iy;

/* 配列 a、b を宣言する。 */
float a[imax], b[imax];

float aa, s, ave;

/* ファイルポインタ fp を宣言する。 */
FILE *fp;

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"r"(テキストファイルの読みこみ)を指定する。 */
fp = fopen( "data05.txt", "r" );

/* ここからデータを読みこむための for ループが始まる。
   データの個数だけ反復する。 */
for (i=0; i<=imax-1; i++)
```

```

{
/* 関数 fscanf でデータを読みこむ。
   ファイルポインタ fp を指定する。
   書式は"%d %f"(整数、浮動小数点)を指定する。
   データファイルの各行の第1要素が年、
   第2要素が配列 a に読みこむデータである。
   関数から変数の値を返すときは、変数の値そのもの(たとえば iy)ではなく、
   その変数へのポインタ(たとえば&iy)が返ってくることに注意。 */
fscanf( fp, "%d %f", &iy, &aa );
a[i] = aa;

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

/* 変数 s にゼロを代入する。 */
s = 0.;

/* ここから for ループが始まる。 */
for (i=0; i<=imax-1; i++)
{
/* 変数 s に a[i]の値を加える。 */
s = s + a[i];

/* ここで for ループが終了する。 */
}

/* 平均値を計算する。 */
ave = s / imax;

/* ここから for ループが始まる。 */
for (i=0; i<=imax-1; i++)
{
/* a[i]と平均値との差を計算する。 */
b[i] = a[i] - ave;
}
}

```

```

/* ここで for ループが終了する。 */
}

/* 結果を書き出す。 */

/* ここからデータを書き出すための for ループが始まる。
   データの個数だけ反復する。 */
for (i=1; i<=imax; i++)
{

/* b[i]の値を標準出力に書き出す。
   書式は" %4d %5.1f"を指定する。
   "%4d"は4ケタの整数、
   "%5.1f"は全体が5ケタで小数点以下が1ケタの浮動小数点という意味である。 */
printf( " %4d %5.1f\n", 1990+i, b[i-1] );

/* ここで for ループが終了する。 */
}

return 0;
}

```

実行例：

```

/home/snaoki> f77 prog05_2.f
/home/snaoki> ./a.out
1991  0.0
1992  0.5
1993 -0.1
<中略>
2010  0.7

```

計算結果を標準出力ではなくファイルに書き出すこともできます。FORTRAN の場合、データを読みこむ場合と同様に OPEN 文を使ってファイルを開きます。それぞれの値を書き出すときは、標準出力の場合と同様に WRITE 文を使います。

プログラム：

FORTRAN

```

C PARAMETER文で定数IMAXを宣言する。
C 定数IMAXはデータの個数を表す。

```


PARAMETER (IMAX=20)

- C 配列A、Bを宣言する。
REAL A(IMAX), B(IMAX)
- C データを読みこむ。
- C ファイルを開く。
- C 機番は10以降の番号を指定する。
- C STATUSは読みこみの場合は' OLD' を指定する。
- C FORMはテキストファイルの場合は' FORMATTED' を指定する。
OPEN(10, FILE=' data05. txt' , STATUS=' OLD' ,
+ FORM=' FORMATTED')
- C ここからデータを読みこむためのDOループが始まる。
- C データの個数だけ反復する。
DO 11 I=1, IMAX
- C READ文でデータを読みこむ。
- C 機番10を指定する。*は書式を指定しないことを示す。
- C データファイルの各行の第1要素が年、
- C 第2要素が配列Aに読みこむデータである。
READ(10, *) IY, A(I)
- C ここでDOループ11が終了する。
11 CONTINUE
- C ファイルを閉じる。
CLOSE(10)
- C 変数Sにゼロを代入する。
S = 0.
- C ここからDOループ21が始まる。
DO 21 I=1, IMAX
- C 変数SにA(I)の値を加える。
S = S + A(I)
- C ここでDOループ21が終了する。

21 CONTINUE

C 平均値を計算する。

C 関数REALで整数型の変数を実数型に変換する。

```
AVE = S / REAL(IMAX)
```

C ここからD0ループ31が始まる。

```
DO 31 I=1, IMAX
```

C A(I)と平均値との差を計算する。

```
B(I) = A(I) - AVE
```

C ここでD0ループ31が終了する。

31 CONTINUE

C ファイルを開く。

C 機番は10以降の番号を指定する。

C STATUSは書き出しの場合は' UNKNOWN' を指定する。

C FORMはテキストファイルの場合は' FORMATTED' を指定する。

```
OPEN(10, FILE=' output. txt' , STATUS=' UNKNOWN' ,  
+     FORM=' FORMATTED' )
```

C ここからデータを書き出すためのD0ループが始まる。

C データの個数だけ反復する。

```
DO 41 I=1, IMAX
```

C B(I)の値を機番10で指定されたファイルに書き出す。

C 書式は' (1X, I4, 1X, F5. 1)' を指定する。

C '1X' は1文字の空白、' I4' は4ケタの整数、

C 'F5. 1' は全体が5ケタで小数点以下が1ケタの実数という意味である。

```
WRITE(10, ' (1X, I4, 1X, F5. 1)' ) 1990+I, B(I)
```

C ここでD0ループ41が終了する。

41 CONTINUE

C ファイルを閉じる。

```
CLOSE(10)
```

```
STOP
```

```
END
```

(参考) C

```
#include <stdio.h>

int main(void)
{

/* 変数 imax を宣言する。
   定数 imax はデータの個数を表す。 */
   int imax=20, i, iy;

/* 配列 a、b を宣言する。 */
   float a[imax], b[imax];

   float aa, s, ave;

/* ファイルポインタ fp を宣言する。 */
   FILE *fp;

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"r"(テキストファイルの読みこみ)を指定する。 */
   fp = fopen( "data05.txt", "r" );

/* ここからデータを読みこむための for ループが始まる。
   データの個数だけ反復する。 */
   for (i=0; i<=imax-1; i++)
   {

/* 関数 fscanff でデータを読みこむ。
   ファイルポインタ fp を指定する。
   書式は"%d %f"(整数、浮動小数点)を指定する。
   データファイルの各行の第1要素が年、
   第2要素が配列 a に読みこむデータである。
   関数から変数の値を返すときは、変数の値そのもの(たとえば iy)ではなく、
   その変数へのポインタ(たとえば&iy)が返ってくることに注意。 */
   fscanff( fp, "%d %f", &iy, &aa );
   a[i] = aa;
```

```

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

/* 変数 s にゼロを代入する。 */
s = 0.;

/* ここから for ループが始まる。 */
for (i=0; i<=imax-1; i++)
{

/* 変数 s に a[i]の値を加える。 */
s = s + a[i];

/* ここで for ループが終了する。 */
}

/* 平均値を計算する。 */
ave = s / imax;

/* ここから for ループが始まる。 */
for (i=0; i<=imax-1; i++)
{

/* a[i]と平均値との差を計算する。 */
b[i] = a[i] - ave;

/* ここで for ループが終了する。 */
}

/* 結果を書き出す。 */

/* ファイルを開く。
   モードは"w"(テキストファイルへの書き出し)を指定する。 */
fp = fopen( "output.txt", "w" );

/* ここからデータを書き出すための for ループが始まる。
   データの個数だけ反復する。 */

```

```

for (i=1; i<=imax; i++)
{
/* b[i]の値をファイルポインタ fp で指定されたファイルに書き出す。
書式は" %4d %5.1f"を指定する。
"%4d"は4ケタの整数、
"%5.1f"は全体が5ケタで小数点以下が1ケタの浮動小数点という意味である。 */
fprintf( fp, " %4d %5.1f¥n", 1990+i, b[i-1] );

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

return 0;
}

```

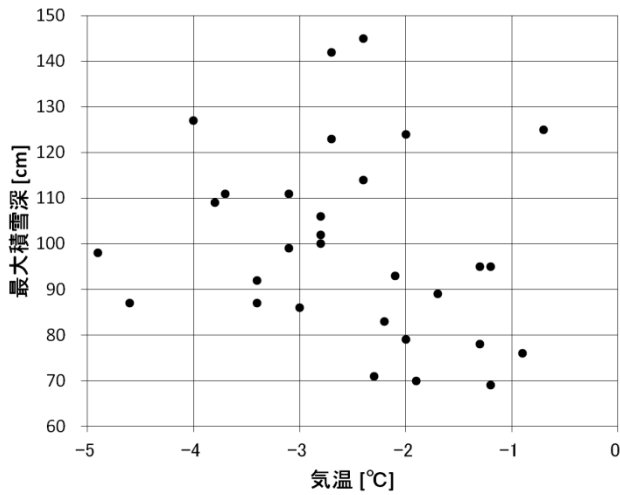
実行例：

```

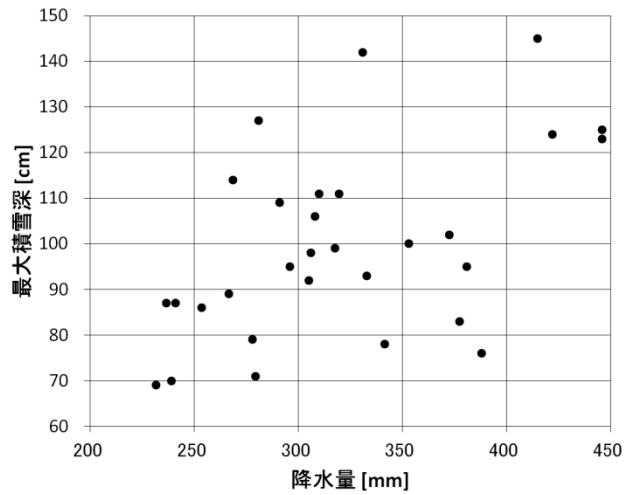
/home/snaoki> f77 prog05_3.f
/home/snaoki> ./a.out
/home/snaoki> cat output.txt
1991  0.0
1992  0.5
1993 -0.1
<中略>
2010  0.7

```

課題5：データファイル temp_sapporo.txt、prec_sapporo.txt、snow_sapporo.txt は、それぞれ 1981～2010 年の札幌における冬季（前年 1 2 月～2 月）の気温、降水量、最大積雪深（気象庁による観測値）である。これらのデータを読みこんで、気温と最大積雪深との間の相関係数と、降水量と最大積雪深との間の相関係数を計算し、小数第 3 位までを標準出力に書きだすプログラムを作成せよ (report05_1.f[.c]、report05_2.f[.c])。余裕があれば、同様の解析を新潟のデータファイル(temp_niigata.txt、prec_niigata.txt、snow_niigata.txt) についても試みよ（ファイルを開くときのファイル名の指定を変えるだけでできるはずである）（新潟については提出不要）。



気温と最大積雪深（札幌）



降水量と最大積雪深（札幌）

（参考）相関係数

相関係数は2つの確率変数の関係性を定量的に表す統計学的な指標である。相関係数は-1から+1までの値をとる。相関係数が正であるということは、一方の確率変数が大きくなるほど、もう一方の確率変数も大きくなる傾向があることを示している。負の場合はこの逆であり、ゼロの場合は相関がないということになる。

2組の数値からなるデータ列 (x_i, y_i) ($i=1, 2, \dots, n$)における x と y の相関係数 r は、

$$r = \frac{\sum_{i=1}^n \{(x_i - \bar{x})(y_i - \bar{y})\}}{\sqrt{\sum_{i=1}^n \{(x_i - \bar{x})^2\}} \sqrt{\sum_{i=1}^n \{(y_i - \bar{y})^2\}}}$$

で定義される。ただし、

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i}{n}$$