第 5 章 演習課題

課題 PDF ダウンロード

課題 1

サンプルプログラムをコンパイル・実行して動作を確認せよ．さらに，適宜修正してその実行結果を確認せよ．

課題 2

与えられた月日 (例えば 4 月 1 日であれば 4 と 1) を標準入力から読み込み，その日が 1 年のうちで何日目かを表示するプログラムを作成せよ．ただし閏年は無視して考えて良い．以下のような配列を用いるとよいだろう．

```
integer, parameter :: days(12) = &
     & (/31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/)
```

実行結果は例えば以下のようなものになる．

```
$ ./a.out
 Input month and day :
4    # キーボード入力
1    # キーボード入力
 day of year :           91
```

課題 3

学生のテストの点数を自動的に処理するプログラムを作成せよ．すなわち，標準入力から学生の人数および人数分のテストの点を順に読み込み，最高点，最低点，平均点，標準偏差をそれぞれ表示するプログラムを作成せよ．ただし標準偏差はデータ数 N ，各データの値 $x_i (i = 1, \ldots, N)$ ，平均値 $\bar{x}$ を用いて

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

と定義される．

データファイル score1.dat を手元にコピーして，以下のようにリダイレクトによって作成したプログラムに読み込ませ，結果を確認せよ．実行結果は例えば以下のようなものになる．

```
$ ./a.out < score1.dat
 Best                :           98
 Worst               :            6
 Average             :    46.399999999999999
 Standard deviation  :    25.115201240152015
```

なおデータファイルには，1 行目にデータ数 N ，それ以降に各データ x_i が記述されているので，まずはデータ数を読み込み配列のメモリを allocate した後に各データを読み込めば良い．(sample3.f90 <chap05_sample3_f90> を参照せよ．)

課題 4

標準入力から 2 つのベクトルを読み込み，両者の内積を計算し表示するプログラムを作成せよ．do ループを用いて地道に計算した結果と組込み関数 dot_product を用いた結果を比較すること．

以下はデータファイル vector.dat を入力とした場合の結果である．

```
$ ./a.out < vector.dat
 Inner product with do loop      :    5.4454054113084460E-017
 Inner product with dot_product :    9.8770817913429454E-017
```

ただしデータは，ベクトルの長さ N ，1 つ目のベクトルの要素 (N 個)，2 つ目のベクトルの各要素 (N 個)，の順に並んでいるものとする．

課題 5

標準入力からベクトルと行列を読み込み，積を計算して表示するプログラムを作成せよ．これについても 2 重 do ループを用いて地道に計算した結果と，組込み関数 matmul を用いた結果を比較すること．

以下はデータファイル matvec.dat を入力とした場合の結果である．これと同じ結果が得られることを確認せよ．

```
$ /a.out < matvec.dat
 Matrix-vector product with do loop
 -0.10000000000000001
 -0.89999999999999991
 -0.50000000000000000
  0.50000000000000000
 -1.5000000000000000
  1.5000000000000000
  1.2000000000000000
 -2.3999999999999999
 Matrix-vector product with matmul
 -0.10000000000000001
 -0.89999999999999991
 -0.50000000000000000
  0.50000000000000000
 -1.5000000000000000
  1.5000000000000000
  1.2000000000000002
 -2.3999999999999999
```

データは，ベクトルの長さ N ，ベクトルの要素 (N 個)，行列の各要素 (N^2 個)，が順に並んでいるものとする．また行列の要素は $a_{11}, a_{21}, a_{31} \ldots$ の順に読み込まれることと，ベクトルと行列の積 $b_i = \sum_j a_{i,j} x_j$ の添字の順番に注意せよ．

matvec.dat は以下のようなファイルになっているが，行列の部分を Fortran で読み込むとあたかも転置行列を読み込んだような形になることに注意せよ．(実際に読み込んで確かめてみよ．)

```
$ cat matvec.dat
 8

   0.1
   1.0
   1.0
   0.5
   0.5
  -1.0
  -1.0
   0.2

  -1.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
   1.0   -2.0    1.0    0.0    0.0    0.0    0.0    0.0
   0.0    1.0   -2.0    1.0    0.0    0.0    0.0    0.0
   0.0    0.0    1.0   -2.0    1.0    0.0    0.0    0.0
   0.0    0.0    0.0    1.0   -2.0    1.0    0.0    0.0
   0.0    0.0    0.0    0.0    1.0   -2.0    1.0    0.0
   0.0    0.0    0.0    0.0    0.0    1.0   -2.0    1.0
   0.0    0.0    0.0    0.0    0.0    0.0    2.0   -2.0
```

Note

プログラムの入力は数学的な「ベクトル」や「行列」を読んでいる訳ではなく，単なる数値の羅列を決められた順番で読み込む．それをどのように「ベクトル」や「行列」として解釈するのかはプログラムを書く人間が決めることである．(配列の入出力 <c5_array_io> を理解するまで熟読せよ．)

課題 6

標準入力から与えられた整数 $n(\geq 2)$ 以下の全ての素数 (1 は素数に含めない) を表示するプログラムを作成せよ．以下のエラトステネスのふるいと呼ばれるアルゴリズムを用いるとよい．

各整数 $i = 2, \ldots, n$ について順に

> ▷ i が素数でなければ無視 ($i+1$ の処理へ)
> ▷ i が素数であれば i から n の整数のうち i の倍数のものを消去 (素数以外と判定)

の処理を行う．なお各整数が素数かどうかを判定するには長さ n の論理型配列を用いれば良い．この配列を全て .true. に初期化し，素数でないと判定されたものは .false. を代入して消去する．

実行結果は例えば以下のようなものになる．

```
$ ./a.out
30                              # キーボード入力
 prime number :            2
 prime number :            3
 prime number :            5
 prime number :            7
 prime number :           11
 prime number :           13
 prime number :           17
 prime number :           19
 prime number :           23
 prime number :           29
```

課題 7

標準入力から 3 つの整数 L, M, N を読み込み，形状が (L, M, N) の整数型の 3 次元配列，および長さ L*M*N の整数型の 1 次元配列を作成せよ．その上で，

> ▷ 組み込み関数 size, shape, lbound, ubound の引数に上記の 2 つの配列をそれぞれ与えた結果を出力し，その動作を確認せよ．
> ▷ reshape を用いて 1 次元配列の中身を 3 次元配列にコピーできることを確認せよ．（ここで 1 次元配列に適当な値を代入してからコピーすることで reshape の動作を確認することもできる．）

```
$ ./a.out
 Input three positive integers (L, M, N) :
2, 5, 7
 --- 3D array ---
 size   (should be equal to L*M*N)               :           70
 shape  (should be equal to 1D array (/L, M, N/) :            2           5           7
 lbound (should be equal to 1D array (/1, 1, 1/) :            1           1           1
 ubound (should be equal to 1D array (/L, M, N/) :            2           5           7
 --- 1D array ---
 size   (should be equal to L*M*N)               :           70
 shape  (should be equal to 1D array (/L*M*N/)   :           70
 lbound (should be equal to 1D array (/1/)       :            1
 ubound (should be equal to 1D array (/L*M*N/)   :           70
```

なお， L, M, N の値はそれぞれせいぜい 100 程度かそれ以下にしておいた方が良い．