CY40014 Introduction to Computational Chemistry
Autumn 2010-2011

Module 1: Introduction to FORTRAN programming
Worksheet 4

Aim of the present worksheet

1. To learn **formatted read and write statements**
2. To learn the use of **character variables**
3. To learn the use of **two-dimensional arrays** to represent **matrices**.

Part 1: Formatted read/write statements

Let us look back at the worksheet problem W3_10 where you were supposed to obtain as output the grades of the students entered against their roll numbers. The instruction that you have so far used for this purpose is:

write(*,*)i,grade(i)

where * indicates that the code is to produce an output free of any format. However, it is possible to specify in the code how the output should be laid out. Some examples are cited below.

Variable type	Statement in code	Comment
Real	write(*,66)x 66 format(f8.3)	8 positions reserved, out of which 3 are for the fractional part; e.g. 3.500
Integer	write(*,73)it 73 format(i6)	6 positions reserved
Real	write(*,66)x,y 66 format(e12.4,1x,e12.4) **OR** 66 format(2e12.4)	12 positions reserved of which 4 are for the E-part as shown below: 0.3456E+01 0.7567E+01 ——————————— 0.3456E+01 0.7567E+01
Real	write(*,66)x,y 66 format(d12.4,1x,d12.4)	Same as e-format, but for double precision
Real	write(*,148)x,y 148 format(e12.4,/,e12.4)	Enters two values in e-format with one in each line 0.3456E+01 0.9876E+01
Character	write(*,9)amino 9 format(a3)	GLU

Exercise:

W4_1. Write a program to read in different types of variables and test the output as given by the use of different formats.

Part 2: Example of usage of character variables

FORTRAN, although developed mainly for formula translation, can also handle **characters** (one or more letters in a string).

- **write(*,*)'Hello world'** : This statement instructs the system to print to the standard output the character string appearing within the quotes.
- It is possible to handle symbol manipulation using FORTRAN.
- Declaring a character variable: **character*4 name**
- In this declaration statement name is a character variable that is comprised of 4 letters. If name='ATOM', it is stored in the following way:

A	T	O	M

- In the case where name='GLU ', the character string will be stored left justified in the memory location and padded with a blank space on the right.

G	L	U	

W4_2. Write a program to store the names of three of your friends in a file. A representative example is given here. This has been written with the assumption that no name is more than 10 letters long.

Note: The number '9' is a statement label. Such labels may be used for reference. For example, '9' represents the reference for the format in which the variable 'name1' is read/written.

The statement labels may be entered between the columns 2-5.

```
      implicit real*4 (a-h, o-z)
      character*10 name1,name2,name3

      write(*,*)'give the names of 3 friends, one per line'
      read(*,9)name1
      read(*,9)name2
      read(*,9)name3
9     format(a10)

      open(unit=8,file='friends.out',status='unknown')
      write(8,9)name1
      write(8,9)name2
      write(8,9)name3
      close(8)

      stop                                    W4_2
      end
```

W4_3. It is possible to use an array for a collection of characters. Write a program that

(i) reads in the 3-letter codes of 20 naturally occurring amino acids from a file 'amino.in'

```
character*3  amino
dimension amino(20)

do i=1,20
    read(12,*)amino(i)
end do
```

(ii) writes the codes to the standard output.
A part of the program is shown on the right.

W4_4. It is also possible to extract a part of the character string. Let us try to find out those amino acids having their three-letter codes starting with 'G'. For this purpose, write a program as in **W4_3** with an additional segment as shown here.

```
character*1 aa

aa=amino(i)(1:1)

if(aa.eq.'G') write(*,*)aa
```

W4_5. First prepare an input file named 'amino2.in' as shown here. Evidently the second column has 0 if the amino acid sidechain is hydrophobic, 1 if it is hydrophilic and 2 if it is amphipathic. Write a program

(i) to read 'amino.in2'
(ii) write the names of hydrophobic, hydrophilic and amphipathic amino acids in separate files named 'hydrophobic.out', 'hydrophilic.out' and 'amphipathic.out', respectively.

```
ALA   0
ARG   1
TRP   2
…
…
….
```

W4_6. Consider the 3 amino acid residues: Phe, Arg and Trp. Read their 1-letter codes from a file named 'amino3.in'. Write a program to arrange these three names in alphabetical order.

[**Note**: This is an interesting point about how the alphabets are manipulated internally by the computers. Most computers code alphabetic characters in creasing order from 'A' to 'Z'. Therefore, 'C' is *less than* 'T'.]

Part 3: An exercise on the use of two dimensional array to represent matrices

An *(n x n)* matrix can be represented in a FORTRAN program using a two dimensional array. While declaring the dimension of the array variable, therefore a matrix ***A*** has to be represented in the following way:

dimension a (100,100)

By convention, the first index on an element a_{ij} of the matrix ***A*** denotes its row and the second index its column. Let ***A*** be a (MxN) matrix. While using FORTRAN code, the matrix is stored by columns in the order

$a_{11}, a_{21}, \ldots\ldots\ldots a_{M1}, a_{12}, a_{22}, \ldots\ldots\ldots, a_{M2}, \ldots\ldots\ldots\ldots, a_{1N}, a_{2N}, \ldots\ldots\ldots, a_{MN}$

However, we are generally used to reading the matrix row wise as shown below.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

In all the problems given below, you are encouraged to save each matrix in a row-wise manner in an input file and use the segment of program shown on the right to read the elements of this matrix in your program.

Note:

- The matrix element you represent as a_{11} in paper becomes *a(1,1)* in your code.
- We are reading the matrix row-wise using an **implicit do loop.**
- It is possible to read the elements of more than one matrices from the same input file.

W4_7. Write a program that reads the elements of the following two matrices:

$$A = \begin{pmatrix} 3 & 1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 2 \end{pmatrix} \quad and \quad B = \begin{pmatrix} 2 & 1 & 1 \\ -1 & 3 & -1 \\ 4 & 1 & 0 \end{pmatrix}$$

and computes

(a) $C = A + B,$ *i.e.* $C_{ij} = A_{ij} + B_{ij}, \quad i, j = 1,2,3$

(b) $D = A - B,$ *i.e.* $D_{ij} = A_{ij} - B_{ij}, \quad i, j = 1,2,3$

(c) $E = AB,$ *i.e.* $E_{ij} = \sum_{k=1}^{3} A_{ik} B_{kj}, \quad i, j = 1,2,3$

A part of the code for addition of two matrices is shown on the right.

For the matrix multiplication, use the program segment shown separately.

```
dimension a(3,3), b(3,3),c(3,3)

do i=1,3
   read(12,*)(a(i,j), j=1,3)
end do
 do i=1,3
   read(12,*)(b(i,j), j=1,3)
end do

 do j=1,3
   do i=1,3
      c(i,j)=a(i,j)+b(i,j)
       write(*,*)c(i,j)
   end do
end do
```

```
do i=1,3
  do j=1,3
     sij=0.0
     do i=1,3
          term=a(i,k)*b(k,j)
          sij=sij+term
      end do
      e(i,j)=sij
   end do
end do
```

W4_8. Write a program that

(a) reads the elements of a matrix A from an input file and

(b) gives in standard output elements of the matrix $B = A^T (B_{ij} = A_{ji})$.

W4_9. Write a FORTRAN program to identify if the matrix A is symmetric or skew symmetric. $A = \begin{pmatrix} 0 & 2 & -3 \\ -2 & 0 & 4 \\ 3 & -4 & 0 \end{pmatrix}$