CY40014 Introduction to Computational Chemistry
Autumn 2010-2011

Module 1: Introduction to FORTRAN programming

Worksheet 3

Aim of the present worksheet

1. To learn the use of subroutines;
2. To learn the use of multiple conditions in IF-construct

Using a subroutine

Let us look at this example where a subroutine is being used to perform a repetitive task, for example, evaluation of a function of *x* for a given value of *x*.

As before, the new programming parts are highlighted in red. The use of a subroutine has been shown in panel (a) while the same program without using the subroutine has been shown in panel (b).

(a)

```
implicit real*4 (a-h, o-z)

read(*,*)x

call getfx(x,fx)
write(*,*)x,fx

stop
end


subroutine getfx(y,fy)
implicit real*4 (a-h, o-z)

fy=y**2

return
end
```

(b)

```
implicit real*4 (a-h, o-z)

read(*,*)x

fx=x**2
write(*,*)x,fx

stop
end
```

It may seem to you that the use of subroutines is quite unnecessary at this stage. You are absolutely right as far as the given example is concerned. However, if you are writing a long and complicated program, subroutines can be used to simplify your task enormously.

Basics of a subroutine

- Subroutines are used to simplify the implementation of a complex computational task in a program.
- A program can use one or more subroutines.
- A subroutine is complete and independent program which can be used (or **call**ed) by the main program.
- A typical way such subroutines are used as shown here.
- In a complicated program, one subroutine can call another subroutine.
- The subroutine receives values (**arguments**) from the calling (or originating) program and then sends back (**return**s) the result(s) to the calling program.

```
implicit real*4 (a-h, o-z)

read(*,*)x

call getfx(x,fx)
write(*,*)x,fx

stop
end


subroutine getfx(y,fy)
implicit real*4 (a-h, o-z)

fy=y**2

return
end
```

Note

You need to declare first if a segment of the program is meant to be a subroutine.

A subroutine does not stop, but **returns** to the main program.

In the example shown, the argument **y** is an input and **fy** is a result that is sent back by the subroutine to the main program.

As far as the representation of the argument is concerned, you need not use the same symbol. But the **same number of arguments** (having the same meaning) MUST be used

(i) while calling the subroutine and

(ii) while declaring the subprogram as a subroutine.

In all the problems given below, you are encouraged to use a subroutine to perform a repetitive task for different sets of data.

W3_1. Write a program that

(i) reads the value of a real variable *x* from standard input,
(ii) evalutes the square of this variable using a subroutine and
(iii) writes the result to the standard output.

W3_2. Write a program that uses the Newton quotient $\frac{f(x+h)-f(x)}{h}$ to estimate the first derivative of *f(x)* at a given value of *x*. Use a subroutine to compute the value of *f(x).*

W3_3. Write a program that

(i) reads *n* values of a real variable *x* from a file,
(ii) evalutes the square of these variables using a subroutine and
(iii) writes the results to the standard output.

A sample program has been shown here on the right.

```
implicit real*4 (a-h, o-z)
dimension x(100)

read(12,*)n
do i=1,n
    read(12,*)x(i)
end do

do i=1,n
   call calfunc(x(i),fxi)
   write(*,*)x(i),fxi
end do

stop
end

subroutine calfunc(y,fy)
implicit real*4 (a-h, o-z)

fy=y**2
                    W3_1
return
end
```

W3_4. A set of *n* integers $\{m_i\}$ is given where *n* is an even number. Write a program that

(i) reads in these *n* integers from a file and
(ii) writes to the standard output the following *n/2* numbers:
$(m_1+m_2),(m_3+m_4),(m_5+m_6),.....(m_{n-1}+m_n)$

A part of the program has been shown below.

```
do i=1,n-1,2
    call sumcal(x(i),x(i+1),sum)
    write(*,*)sum
end do
```

W3_5. An array can be exchanged between the main program and the subroutine. Here is an application to demonstrate how to carry out such exchanges. Suppose you are given the two vectors $\vec{A}$ and $\vec{B}$ in the three- dimensional space. Write a program that

(i) reads in the components of these two vectors from a file,
(ii) calculates their sum.

A part of the program has been shown on the right. Complete the program, compile and run it.

Note: *If you are using the same array variable in the main program and subroutine, make sure that its dimension is declared in both the places.*

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

call vecsum(a,b,sumab)

write(*,*)'sum of two vectors'
do i=1,3
    write(*,*)sumab(i)
end do

stop
end

subroutine sumab(a,b,s)
implicit real*4 (a-h, o-z)
dimension a(3),b(3),s(3)

do i=1,3
    s(i)=a(i)+b(i)
end do
return
end
```

W3_6. Given the two vectors $\vec{A}$ and $\vec{B}$ in the three- dimensional space. Write a program that

(i) reads in the components of these two vectors from a file,
(ii) calculates the corresponding difference vector.

W3_7. Given the two vectors $\vec{A}$ and $\vec{B}$ in the three- dimensional space. Write a program that

(i) reads in the components of these two vectors from a file,
(ii) calculates the corresponding scalar product.

W3_8. Given the two vectors $\vec{A}$ and $\vec{B}$ in the three- dimensional space. Write a program that

(i) reads in the components of these two vectors from a file,
(ii) calculates the corresponding summation and difference vectors and also the scalar product.

Note that in this program you will be using three subroutines along with the main program.

A useful tip

Getting into trouble with input file names? Here is a useful hint to make your life simpler. If you are solving worksheet problem W3_2, name your program file as **w3_2.f** and the corresponding input file **w3_2.in**. The output file, if any, may be named as **w3_2.out**.

Simple, don't you think so?

Exercises on the use of if statements with multiple decisions

W3_9. A student has scored 78% marks in his final examination. Write a program to determine his grade according to the following list.

% of marks	Grade
0-35	F
36-50	P
51-100	A

```
write(*,*)'give percentage scored'
read(*,*)x

if(x.le.35.0) then
   write(*,*)'grade = F'
endif

if((x.gt.35).and.(x.le.50.0)) then
   write(*,*)'grade = P'
endif

if(x.gt.50.0) then                    W3_9
  write(*,*)'Grade = A'
endif
```

W3_10. There are 10 students in the class who have scored as follows:

Roll no.	1	2	3	4	5	6	7	8	9	10
%	92	49	88	6	27	35	51	76	97	5

Write a program that writes to the standard output the grades obtained by these students according to the rule shown in W3_9.

Assignment

A3_1. Consult any textbook and read about how to write and use a **function subprogram**.

A3_2. Write a program that uses the Newton quotient $\frac{f(x+h)-f(x)}{h}$ to estimate the first derivative of *f(x)* at a given value of *x*. Use a function subprogram to compute the value of *f(x)*.

A3_3. Compute the value of *sinh(x)* for *10* different values of *x*. Use a function subprogram to calculate *exp(x)* and then use it to find *sinh(x)*.

A3_4. Write a Fortran program to print all the Fibonacci numbers less than 50. (Hint: The Fibonacci numbers are 0,1,1,2,3,5,8,13,……..). Start by expressing the *n*-th Fibonacci number F_n from F_{n-1} and F_{n-2}.

A3_5. The bonus to be paid in a company is calculated as per the following rule:

Grade of employee	Bonus
1	75% of one month salary
2	90% of one month salary subject to a maximum bonus of Rs.2500/_
3	One month salary and an additional amount of Rs.200/_ subject to a maximum bonus of Rs.2000/_

It is also known that nobody can be paid a bonus of more than Rs. 3500/_. Write a Fortran program that reads in an employee's grade and salary and gives the bonus due as output.

A3_6. From a given set of numbers $x_1, x_2, x_3, ..., x_N$, write a program to print all the *distinct* pairs of numbers. Also, write in output the total number of such pairs possible. [*Hint: for the numbers* x_1, x_2 *and* x_3, *there are three distinct pairs, viz.* (x_1, x_2), (x_1, x_3) *and* (x_2, x_3). *Note that the pairs* (x_1, x_2) *and* (x_2, x_1) *are identical for all practical purposes*.]

A3_7. From a given set of numbers $x_1, x_2, x_3, ..., x_N$, write a program to print all the *distinct* triplets of numbers. Also, write in output the total number of such triplets possible.

A3_8. A set of *15* integers $\{m_i\}$ is given. Write a program that

(i) reads in these integers,

(ii) forms triplets as $(m_1, m_2, m_3), (m_4, m_5, m_6), ...$

(iii) calculates the products $(m_1 m_2 m_3), (m_4 m_5 m_6), ...$

(iv) writes to the standard output the largest product **and the three numbers that gave this product.**

A3_9. Write a program to compute the roots of the quadratic equation $ax^2 + bx + c = 0$. Include a subroutine that returns to the main program

(i) an integer *j* that takes up the following values:

(ii) the roots if they are real.

***j* = -1 : complex roots (discriminant <0)**
= 0 : no solution ($a = b = 0, c \neq 0$)
= 1 : one root ($a = 0, b \neq 0$)
= 2 : two roots (which could be equal)
= 99 : any *x* is a solution ($a = b = c = 0$)

Note The problems given in blue are optional.

A3_10. A set of *n* integers $\{m_i\}$ is given where *n* is an even number. Write a program that

(i) reads in these *n* integers from a file,

(ii) forms *n/2* pairs as $(m_1, m_2), (m_3, m_4), (m_5, m_6), \ldots\ldots (m_{n-1}, m_n)$ and

(iii) writes to the standard output the smaller one of each pair thus formed.