前提：

Xterm(X端末)上でUNIXコマンドを学び，vi editorまで使えるとする。(わからなければ本に戻ればよい。)

mac上のxtermでコマンドで以下をたたく

gfortran -v ↵

以下のような出力が出ればOK→gfortran が動いている

```
masahiromacproxeonqcore-3:~ masahiroyamamoto$ gfortran -v
Using built-in specs.
COLLECT_GCC=gfortran
COLLECT_LTO_WRAPPER=/usr/local/libexec/gcc/x86_64-apple-darwin10/4.6.0/lto-wrapper
Target: x86_64-apple-darwin10
Configured with: ../gcc-4.6-20100703/configure --enable-languages=fortran --host=x86_64-apple-darwin10 --
build=x86_64-apple-darwin10
Thread model: posix
gcc version 4.6.0 20100703 (experimental) (GCC)
```

a=10.0, b=3.0, c=a+b　を計算します。

test.fというファイル（以後プログラム）をつくります。

(プログラム解説は次頁)

vi test.f　↵

-------ここから（この行必要無し）以下カラムもそろえて！２行目以降は7カラムから記述すること

```
c1234567890
              implicit real*8 (a-h,o-z)
              a=10.0d0
              b=3.0d0
              c=a+b
              write (6,*) a,b,c
              stop
              end
```

----------ここまで（この行必要無し）

vi 画面上のコントロールモードで**zz**でファイル保存

プログラム解説

```
c1234567890 !プログラムは7カラム目から入力します その確認のための行
      implicit real*8 (a-h,o-z) !a-h,o-zで始まる変数は16桁の倍精度です
      a=10.0d0 ! **d0は10.0×10^0の倍精度,　1000は1.0d3あるいは1000.0d0とかける
      b=3.0d0
      c=a+b    ! +足し算, -引き算, ×かけ算 , /割り算,  べき乗** (√2は2.0**0.5d0)
      write (6,*) a,b,c  !a,b,cの結果を6番（画面)に出力する. *は任意の型式で
      stop          !プログラムの実行はここでおわり
      end           ! プログラムはここでおわり
```

プログラムの機械語への翻訳:コンパイル

gfortran tes.f ↵

プログラムにエラーがなければ実行可能なファイル
a.outがプログラムが存在するdirectoryに作られる。

プログラムの実行

a.out ↵

その結果，以下の様な画面が表れればOK

```
masahiromacproxeonqcore-3:~ masahiroyamamoto$ a.out
   10.000000000000000        3.0000000000000000        13.000000000000000
```

1.0から100.0までを加算する：ループ計算

test2.fというプログラムをつくります。

vi test2.f ↵

----------ここから（この行必要無し）色は入力必要無し

```
c234567890
        implicit real*8 (a-h,o-z)
        sum=0.0d0     !初期値入力

        do i=1, 100 !ループ青の部分を100回くりかえします。
          a=real(i)    !iは整数なので実数にします.real( )は実数にする関数
          sum=sum+a    !aをそれまでのsumに加えます
          write (6,*) a,sum  !ループの中は１カラム右（８カラムから）に書きます
        enddo    !ループはここまで

        stop
        end
```

----------ここまで（この行必要無し）

vi 画面上のコントロールモードでzzでファイル保存

翻訳・実行・出力： その結果，以下の様な画面が表れればOK

```
masahiromacproxeonqcore-3:~ masahiroyamamoto$ gfortran test2.f ↵
masahiromacproxeonqcore-3:~ masahiroyamamoto$ a.out ↵
   1.000000000000000       1.000000000000000
   2.000000000000000       3.000000000000000
   3.000000000000000       6.000000000000000
   4.000000000000000       10.00000000000000
   5.000000000000000       15.00000000000000
   6.000000000000000       21.00000000000000
   7.000000000000000       28.00000000000000
   8.000000000000000       36.00000000000000
   9.000000000000000       45.00000000000000
   10.00000000000000       55.00000000000000
...
   90.00000000000000       4095.000000000000
   91.00000000000000       4186.000000000000
   92.00000000000000       4278.000000000000
   93.00000000000000       4371.000000000000
   94.00000000000000       4465.000000000000
   95.00000000000000       4560.000000000000
   96.00000000000000       4656.000000000000
   97.00000000000000       4753.000000000000
   98.00000000000000       4851.000000000000
   99.00000000000000       4950.000000000000
   100.0000000000000       5050.000000000000
```

出力を画面だけではしんどいのでファイルに書き出します。

test3.fというプログラムをつくります。

vi test3.f ↵

----------ここから（この行必要無し）

```
c234567890
      implicit real*8 (a-h,o-z)
      open(1, file='aaa.txt') ! aaa.txtというファイルに書き出す準備
      sum=0.0d0 !初期値入力        ⏎
      do i=1, 100 !赤の部分を100回くりかえします。
       a=real(i)    !iは整数なので実数にします
       sum=sum+a    !aをそれまでのsumに加えます
       write (1,*) a,sum ! 1は'aaa.txt'というテキストファイルです
      enddo
      close(1)！ファイルを閉じます
      stop
      end
```

----------ここまで（この行必要無し）

翻訳・実行・出力

masahiromacproxeonqcore-3:~ masahiroyamamoto$ gfortran test3.f ↵
masahiromacproxeonqcore-3:~ masahiroyamamoto$ a.out ↵

なにも画面には出力されません。

vi aaa.txt ↵

でaaa.txtのファイルを確認して下さい。

これでプログラムで数字データを書くことが出来るようになりました。

ファイルの読み込み・配列を使った計算

次に，先ほど作成した‘aaa.txt’を読み込んでその値および値間の演算を行います。test4.fというファイルをつくります。

vi test4.f ↵

----------ここから（この行必要無し）色の入力はできない

```
c234567890
        implicit real*8 (a-h,o-z)
        integer, parameter : :  ndim=500 !配列の大きさ
        dimension d(ndim),  e(ndim) !配列はd(1),d(2),....,d(500)まで
        open(1, file=‘aaa.txt’)    !入力ファイルの準備
        open(2, file=‘bbb.txt’)    !出力ファイルの準備
        do i=1, 100
         read (1,*) d(i), e(i)
        enddo
        write (6,*) d(1),d(2), d(99), d(100) !ちゃんと読み込んでいるかチェック
        write (6,*) e(1),e(2), e(99), e(100)
        do i=2, 100
           write (2,*) i, d(i)-d(i-1),  e(i)-e(i-1) ! i行,  第一列,  第２列の引き算をファイルbbb.txtに
        enddo
       close(1)
       close(2)
       stop
       end
```

----------ここまで（この行必要無し）

masahiromacproxeonqcore-3:~ masahiroyamamoto$ gfortran test4.f ↵
masahiromacproxeonqcore-3:~ masahiroyamamoto$ a.out ↵

画面には以下の出力

```
 1.0000000000000000       2.0000000000000000       99.000000000000000       100.00000000000000
 1.0000000000000000       3.0000000000000000       4950.0000000000000       5050.0000000000000
```

ファイルbbb.txtは以下のようになる。

vi bbb.txt ↵

```
   2   1.0000000000000000       2.0000000000000000
   3   1.0000000000000000       3.0000000000000000
   4   1.0000000000000000       4.0000000000000000
   5   1.0000000000000000       5.0000000000000000
   6   1.0000000000000000       6.0000000000000000
   7   1.0000000000000000       7.0000000000000000
   8   1.0000000000000000       8.0000000000000000
   9   1.0000000000000000       9.0000000000000000
  10   1.0000000000000000       10.000000000000000
  11   1.0000000000000000       11.000000000000000
  12   1.0000000000000000       12.000000000000000
  13   1.0000000000000000       13.000000000000000
  14   1.0000000000000000       14.000000000000000
  15   1.0000000000000000       15.000000000000000
...
  95   1.0000000000000000       95.000000000000000
  96   1.0000000000000000       96.000000000000000
  97   1.0000000000000000       97.000000000000000
  98   1.0000000000000000       98.000000000000000
  99   1.0000000000000000       99.000000000000000
 100   1.0000000000000000       100.00000000000000
```

非常に簡単な例だけですが，
もうちょっと勉強すれば何でも計算できそうですよね！

プログラムを書けるということは，今後ますます重要になり，ある意味「A層：創造する側」か「B層：ユーザー」かに別れてくると思います。当然A層にいることは何かとアドバンテージがありますよね。

FOTRAN入門の本があります。

培風館　『Ｆｏｒｔｒａｎ９０/９５プログラミング』

（改訂新版）　冨田博之・齋藤泰洋 共著：　2011年４月改訂新版　2013年９月４刷

例題のソースファイル（プログラム）が以下にあります。

http://www7b.biglobe.ne.jp/~fortran/education/fort9095.html

Nagという会社が提供してます「Fortran入門」は以下から読めます

http://www.nag-j.co.jp/fortran/index.html

NagのFortran検定を受けてみましょう

http://www.nag-j.co.jp/fortran/exam/index.html