Examination ProgMod

C. Thieulot and A.P. van den Berg

October 2, 2014

1 Exercise 1

Write a program which

1. (0.5 pt) prompts the user for three real numbers
2. (0.5 pt) computes and prints their sum 's'
3. (0.5 pt) prints 'too big' if their product exceeds 100

2 Exercise 2

1. (1pt) Write a function which takes as argument an integer n and returns $n!$ ("factorial n")
2. (1pt) Write a small program which makes use of this function

3 Exercise 3

1. (1pt) Write a subroutine which takes as arguments an integer n and a real x and computes $\sqrt{x^n}$
2. (1pt) Write a small program which makes use of this subroutine

4 Exercise 4

Write a fortran program according to the following specifications:

1. (0.5 pt) We wish to store in the arrays **xcoords**, **ycoords** and **zcoords** the x, y, z coordinates of a yet unknown number of points. Declare these arrays.
2. (0.5 pt) Have the user choose the size N of the arrays and fill them with random numbers.
3. (1pt) Open a file (name it as you wish) and use a do-loop to write the coordinates of these points on three columns
4. (1pt) compute the following quantities:

$$\alpha = \sum_{i=1}^{N} |x_i| \qquad \beta = \frac{1}{N}\sum_{i=1}^{N}(x_i + y_i) \qquad \delta = \min_{i=1,N} z_i$$

➡

5 Exercise 5

(1.5 pt) The result of squaring a number can also be arrived at by progressively adding consecutive odd numbers as shown below.

1^2	= 1	= 1
2^2	= 4	= 1+3
3^2	= 9	= 1+3+5
4^2	= 16	= 1+3+5+7
5^2	= 25	= 1+3+5+7+9
6^2	= 36	= 1+3+5+7+9+11
7^2	= 49	= 1+3+5+7+9+11+13
8^2	= 64	= 1+3+5+7+9+11+13+15
9^2	= 81	= 1+3+5+7+9+11+13+15+17
10^2	= 100	= 1+3+5+7+9+11+13+15+17+19

n^2 i= 1,n

17= 2*9 - 1

```
if ((0/= mod(i,2)) then
  summation(i)= sum(i) + 1
end ife
end if
```

Write a program which verifies this (Note that 17=2*9-1).
The instructions are voluntarily kept to a minimum. Be creative !

Recommendations:

- Every single used variable has to be defined.
- Is it an integer ? a real ? a static array ? an allocatable array ? etc ...
- Comment your code appropriately.
- Points will be deduced for unclear/unreadable statements.
- Every single used variable has to be defined (I insist).

①

```
p.
in.
real abcs
print
read
s=(a+b+c)/3
print s
if((a*b*c)>100) then.
print
else
end if
end p.
```

②

```
p
i.n.
integer, external :: funcn      ←
integer n                       ←
print give n
read n
print funcn (n)                 ←
end p.
! -----------------------
function funcn (n)
i. n.
integer n, i, funcn, f          ←
do i = 1, n
    f = 1. * i
    funcn = sum (f)             ←
end do
end function
```

③

```
program
i.n.
integer n                       ←
real x, c                       ←
print give integer
read n
print give real
read x
call compute (x, n, c)          ←
print c
end prog
! -----------------------
subroutine compute (x, n, c)    ←
i.n.
integer n                       ←
real x, c                       ←
c = (sqrt (x**n)) * 1.
end subroutine
```

4:

```
prog
i. n.
integer, dimension (:),allocatable  ::  xcoords   ⇐
 integer  n
print    hics n
read       n              {real  a, b, d

allocate    xcoords(n)  ⇐
call   random_number (xcoords)  ⇐

open  (unit=111 , file = 'random.dat')
    do   i = 1, n
    write (111,*)    xcoords, ycoords, zcoords
    end do
close (111)

 ~~ex~~ sum=0
 do  i = 1,n
            summation    summation
①    sum= sum(xcoords(i)) + 1
 end do


②  do   i = 1,n
       B1=  x(i)  +  y(i)
    end do
    B2 =  B1 * 1/n

↕ print*,  'a=' , summation ,  'b=' , B2,

③  do  i=1,n
      ~~min~~ d  =  min (zcoords (i))
   end  do
```

```
5   ~~0 /= mod(i,2)~~
    prog
    i.n
    integer   i, n , summ, n2
    print  geef n
    read  n
    summ=0
    if   (0 /= mod (i,2)) then
      do  i = 1,n
      summ(i) = summ(i) + 1
      end do
    else
    end if

    n2 = n**2
    ~~print~~*, n2 , summ
    if  (n2 == summ) then
    print   'equal they are !'
    else
    end if
    end prog.
```