Quiz 3 (Close book, no cheat sheet, 45 minutes)

Student ID_________________________Name (Print)__________________________

Your paper will not be graded unless you endorse the following statement: 
I have neither given nor received inappropriate assistance on this quiz.

Signature___________________________

Fortran keywords that might be useful in this quiz:
program, end, implicit none, integer, real, character, parameter, ::, !, &,
>=, <, <=, ==, /=, .and., .or., .eqv., .neqv., if, then, else, else if,
end if, select, case, default, :, end select, stop,
function, subroutine, call, intent, in, out,
do, end do, do while, exit
dimension, (/,

Part I Multiple Choice (40 points, 4 points each)
Each problem has one correct answer. Clearly write the letter corresponding to the correct
answer in the boxes on the right.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Select the correct Fortran 95/2003 declaration statement for an integer array:</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>A. Real, dimension(10)::array=0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Integer::array=0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Integer,intent(in)::array</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. Integer, dimension(10)::array=10</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>What is the size of the array declared as the following Fortran 95/2003 statement “real, dimension(5, 3)::threeD=0”?</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>A. 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. 15</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>What is the rank of a 3x4x5 array?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>A. 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. 5</td>
<td></td>
</tr>
</tbody>
</table>
4. Which Fortran 95/2003 statement is NOT correct for array initialization?
   A. Character(2), dimension(10)::state='NA'
   B. Logical, dimension(3)::correct=.false.
   C. Integer, dimension(2)::a=/(0, 0, 0)/
   D. Real, dimension(2)::b=/(0.0, 0.0/)
   **C**

5. Which Fortran 95/2003 keyword CANNOT be used in argument declaration?
   A. dimension
   B. intent
   C. parameter
   D. real
   **C**

6. Which statement best summarize the control structure of the following pseudocode?
   ```fortran
   Do i=1,10,2
     Do j=i,10
       Call doSomething(i, j )
     End do
   End do
   ```
   A. loop
   B. nested loop
   C. branching
   D. nested branching
   **B**

7. What will the write statement display given the following partial Fortran code?
   ```fortran
   integer, dimension(3)::arr=(/3,1,2/)
   write(*,*) arr, arr(2)
   ```
   A. 3 1 2 1
   B. 3 2 1 2
   C. arr 1
   D. arr 2
   **A**

8. What will the write statement display given the following partial Fortran code?
   ```fortran
   integer, dimension(3)::arr=(/3,1,2/)
   integer::counter=0
   do counter = 1,3
     arr(counter) = arr(counter)*counter
   end do
   write(*,*) arr
   ```
   A. 3 1 2
   B. 3 2 2
   C. 3 2 4
   D. 3 2 6
   **D**
9. What is the purpose of the “implicit none” statement?
   A. Force compiler to check variable declarations
   B. Force compiler to check data holder types
   C. Force compiler to check function declarations
   D. All of the above

10. What will the write statement display given the following partial Fortran code?

    ```fortran
    integer,dimension(3)::arr=(/3,1,2/)
    integer::i=0,j=0, t=0
    do i = 1,2
        do j = i, 3
            if ( arr(j) < arr(i) ) then
                t = arr(i)
                arr(i) = arr(j)
                arr(j) = t
            end if
        end do
    end do
    write(*,*) arr
    ```

    A. 3 1 2
    B. 3 2 1
    C. 1 2 3
    D. 0 0 0

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>D</td>
</tr>
<tr>
<td>10</td>
<td>C</td>
</tr>
</tbody>
</table>
Part II Interpreting programs (25 points)

11. (10 points) Read the following partial Fortran code:
   ```fortran
   integer, dimension(6):: testLoop=0
   integer:: i=0
   do while ( i <= 6 )
      if ( i <= 4 ) then
         testLoop(i) = i * 2
         i = i + 1
      else
         testLoop(i) = i
         i = i + 2
      end if
   end do
   ```

1) There is a bug in the above code, and you can modify one statement to fix it. Please list the original statement and your bug-fixing version of it:

   Original: __
   ```fortran
   integer:: i=0
   ```

   Fixed: ____
   ```fortran
   integer:: i=1
   ```

2) Please list the values of the array `testLoop` at the end of the loop:

   testLoop(1): ____ 2 _______  testLoop(2): ____ 4 _______
   testLoop(3): ____ 6 _______  testLoop(4): ____ 8 _______
   testLoop(5): ____ 5 _______  testLoop(6): ____ 0 _______
12. (15 points) The following code stored in q3p12.f95 is not compiling, it also has a logic error. Please debug the code according to the guidance in part a) and b). You do not need to read through the code at this point, compiler messages and runtime information are provided to help you debug.

```fortran
! Interpolate a 2D line segment given the coordinate of the two end points of the line, and the Euclidean distance between adjacent points
program lineInterp
implicit none
integer :: MAXSIZE = 1000
real, dimension(MAXSIZE):: x = 0, y = 0
real:: x0=0,y0=0,xn=0,yn=0,minDist=0
real:: dist=0,distX=0
integer::n=0,i=0
!
! get user input
read(*,*) x0,y0,xn,yn
read(*,*) minDist
!
! calculate Euclidean length of the line segment
! validate the number of points
n = floor(dist/minDist+.5)+1
if ( n > MAXSIZE )
write(*,*) "dist of ", minDist, " is too small"
stop
!
! linear interpolation of the points
do i = 1, n
x = x0 + distX*(i-1)
y(i) = y0 + distY*(i-1)
end do
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
! display the results
write(*,*) i, x(i), y(i)
end do
end program
```
a) Please correct syntax errors according to the following compiler message:

```bash
[jxue@backus tests]$ f95 q3p12.f95
In file q3p12.f95:23

    if ( n > MAXSIZE )
    1
Error: Unclassifiable statement in IF-clause at (1)
In file q3p12.f95:26

    end if
    1
Error: Expecting END PROGRAM statement at (1)
In file q3p12.f95:18

    distY = yn-y0
    1
Error: Symbol 'disty' at (1) has no IMPLICIT type
In file q3p12.f95:7

    real, dimension(MAXSIZE)::x = 0, y = 0
    1
Error: Variable 'maxsize' cannot appear in the expression at (1)
In file q3p12.f95:7

    real, dimension(MAXSIZE)::x = 0, y = 0
    1
Error: The module or main program array 'x' at (1) must have constant shape
In file q3p12.f95:7

    real, dimension(MAXSIZE)::x = 0, y = 0
    1
Error: Variable 'maxsize' cannot appear in the expression at (1)
In file q3p12.f95:7

    real, dimension(MAXSIZE)::x = 0, y = 0
    1
Error: The module or main program array 'y' at (1) must have constant shape
```

Syntax error1 at line ___________ 23 ______________

Corrected statement ___if ( n > MAXSIZE ) then___
Syntax error2 at line _________ 9 ___________

Corrected statement ______ real::dist=0,distX=0,distY=0 ___________

Syntax error3 at line _________ 6 ___________

Corrected statement ______ integer, parameter::MAXSIZE = 1000 ___________

b) Now the code can be compiled with no errors, but the result is not correct as shown in the following test:

```
[jxue@backus tests]$ f95 q3p12.f95
[jxue@backus tests]$ ./a.out
0 0
1 1
0.5
   1 1.000000 0.000000
   2 1.000000 0.3333333
   3 1.000000 0.6666667
   4 1.000000 1.000000
[jxue@backus tests]$
```

There is only one place in the line interpolation loop that holds the logic error.

Logic error at line _________ 32 ___________

Corrected statement ____ x(i) = x0 + distX*(i-1) ___________
Part III. Writing a Fortran 95/2003 procedure (35 points, no comments are required).

13. Write a subroutine that collects statistics of an input array including its average, minimum, and maximum. The input array is in real type and with a rank of 1.

a) What are the inputs to the subroutine?

Input argument 1: input array, arr

Input argument 2: size of the input array, N

b) What are the outputs of the subroutine?

Output argument 1: minimum from the input array, minVal

Output argument 2: maximum from the input array, maxVal

Output argument 3: average of the input array, average

c) One big loop construct can get all the output results. Please complete the following pseudo-code for the algorithm:

initialize maxVal
initialize minVal
initialize average
do i = 1, N
    if ( maxVal < arr(i) ) then
        maxVal = arr(i)
    end if
    if ( minVal > arr(i) ) then
        minVal = arr(i)
    end if
    average = average + arr(i)
end do
average = average/N

d) Now you are ready to implement the code in Fortran 95/2003.

Subroutine arrayStatistics( arr, N, minVal, maxVal, average )
Implicit none

    Integer, intent(in) :: N
Real, intent(in) :: arr

Real, intent(out) :: minVal

Real, intent(out) :: maxVal

Real, intent(out) :: average

Integer::i=0

   minVal = arr(1)
   maxVal = arr(1)
   average = 0

Do i = 1, N

   If ( minVal > arr(i) ) then
      minVal = arr(i)
   end if

   If ( maxVal < arr(i) ) then
      maxVal = arr(i)
   end if

   average = average + arr(i)
end do

average = average/N
end subroutine

e) Now write a unit test program to verify your subroutine implementation:

program testArrayStatistic
    implicit none

    real,dimension(4)::a=(/3,4,1,5,0/)
    real::minV=0,maxV=0,ave=0

    call arrayStatistics( a, 4, minV, maxV, ave)

    write(*,*) “min=” , minV, “max=” , maxV, “average=” , ave
end program
Part IV. Optional problem (20 points)

13. Write a function to calculate the Pearson correlation coefficients between two vectors as shown in the following equation:

\[
\rho = \frac{(X - \mu_x)^T(Y - \mu_y)}{\sqrt{(X - \mu_x)^T(X - \mu_x)} \sqrt{(Y - \mu_y)^T(Y - \mu_y)}}
\]

where X and Y are column vectors of equal length, \(\mu_x\) and \(\mu_y\) are the averages of vectors X and Y, respectively. You can use the innerProd function developed in Homework 5. You can also use the subroutine you implemented in Prob12.

```plaintext
real function pearsonCorr(X,Y,N)
  implicit none
  integer, intent(in)::N
  real, intent(in), dimension(n)::X,Y
  integer::i=0
  integer, parameter::MaxSize=1e6
  real::dummy=0.,muX=0.,muY=0.
  real,dimension(MaxSize)::Xn=0,Yn=0
  real::innerProd
  
  if ( N > MaxSize ) then
    pearsonCorr = -2
  else
    call arrayStatistics(X,N,dummy,dummy,muX)
    call arrayStatistics(Y,N,dummy,dummy,muY)
    do i=1,N
      Xn(i)=X(i)-muX
      Yn(i)=Y(i)-muY
    end do
    pearsonCorr = innerProd(Xn,Yn,N)/ &
          sqrt( innerProd(Xn,Xn,N) &
                * innerProd(Yn,Yn,N) )
  end if
end function
```