Quiz 2 (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, cycle

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.

<table>
<thead>
<tr>
<th>1. Which statement can get user input from keyboard?</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. write(<em>,</em>) “Enter an input value”</td>
<td></td>
</tr>
<tr>
<td>B. read(<em>,</em>) value</td>
<td></td>
</tr>
<tr>
<td>C. value = 10</td>
<td></td>
</tr>
<tr>
<td>D. input = 'value'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Can you terminate a program in between two executable statements?</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>If yes, which statement can do the job?</td>
<td></td>
</tr>
<tr>
<td>A. no</td>
<td></td>
</tr>
<tr>
<td>B. yes, exit</td>
<td></td>
</tr>
<tr>
<td>C. yes, stop</td>
<td></td>
</tr>
<tr>
<td>D. yes, quit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A. yes</td>
<td></td>
</tr>
<tr>
<td>B. no</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Which Fortran procedure can be used in a relational logical expression?</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. subroutine</td>
<td></td>
</tr>
<tr>
<td>B. function</td>
<td></td>
</tr>
<tr>
<td>C. program</td>
<td></td>
</tr>
<tr>
<td>D. class</td>
<td></td>
</tr>
</tbody>
</table>
5. Which of the following does NOT need to be declared before being used in an executable statement in Fortran 95/2003?
   A. local variables  
   B. function  
   C. subroutine arguments  
   D. subroutine
   D

6. In a while loop, where inside the loop can you place a block if construct to prevent the loop from infinite iterations?
   A. begin  
   B. middle  
   C. end  
   D. all of the above
   D

7. Which of the following statements can be used to declare an input argument, inside a subroutine named zebra, in Fortran 95/2003?
   A. integer::dog = 0  
   B. real,intent(out)::cat = 0  
   C. logical::zebra  
   D. character, intent(in)::monkey
   D

8. Which statement can be used to round a real typed variable val to its closest integer?
   A. if (val < 1.0) then  
   B. if (val < int(val)+0.5) then  
   C. if (val == “f”) then  
   D. if (abs(val – 1.0) > 1e-10) then
   B

9. Assume the variable index is in integer type, according to the following counting loop statements:
   
   ```fortran
   do index = 5, 10, 2
     call doSomething( index )
   end do
   ```
   How many times will the subroutine doSomething be executed at the end of this loop?
   A. 3  
   B. 4  
   C. 5  
   D. 6
   A

10. Select the proper control construct that can translate the following formula into a Fortran 95/2003 program without calling any subroutine or function:
    
    $$f(m,n) = \begin{cases} 
    1 & \text{if } n = 0 \text{ and } m = 0 \\
    \frac{n!}{m!(n-m)!} & \text{if } n > m \geq 0 
    \end{cases}$$
    A. sequential  
    B. branching  
    C. loop  
    D. loops nested inside branching
    D
Part II Interpreting programs (20 points, 10 points each)

11. Read the following partial Fortran code:
   integer::i=0, foo=0
   do while ( i <= 6 )
     if ( i < 3 ) then
       foo = foo + i * 2
       i = i + 1
     else
       foo = foo + i
       i = i + 2
     end if
   end do

1) What is the value of the variable $i$ at the end of the first iteration?

   1

2) What is the value of the variable $foo$ after the end of the last iteration?

   14
12. Read the following flowchart:

Predict the output of the program given the following input:

<table>
<thead>
<tr>
<th>Input of Val</th>
<th>Monitor Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>Invalid input</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
Part III. Writing a Fortran 95/2003 procedure (40 points, no comments are required).

13. The Riemann integral of a function \( f(x) \) with \( x \) in the interval \([a, b]\) is defined by taking the limit of a Riemann sum as follows:

\[
\int_a^b f(x) \, dx = \lim_{n \to \infty} \sum_{i=0}^{n-1} f\left(\frac{x_{i+1} + x_i}{2}\right) \left(x_{i+1} - x_i\right)
\]

where \( a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b \), and \( n \) is the number of subintervals. The \( i \)th subinterval is defined by \([x_{i-1}, x_i]\). Suppose a function \( f(x) \) has been implemented into a Fortran 95/2003 function \( \text{foo} \) with the following function statement and argument declaration:

```fortran
real function foo( x )
  implicit none
  real, intent(in)::x

end function
```

Please write a Fortran 95/2003 function to calculate the Riemann sum (the highlighted part in the equation) of the above function \( \text{foo} \). Your Riemann sum function should have three input arguments including the interval boundaries \( a \) and \( b \) in real type, and the number of subintervals \( n \) in integer type. You can generate subintervals with identical length. The function should return a real type result of the Riemann sum.

```fortran
real function riemannSum( a, b, n )
  implicit none
  real,intent(in)::a,b
  integer,intent(in)::n

  real::dx=0,x=0
  real::foo

  dx = (b-a)/n
  x = a + dx/2
  riemannSum = 0
  do while ( x < b )
    riemannSum = riemannSum+foo(x)*dx
    x = x + dx
  end do
end function
```
Part IV. Optional problem (20 points)

13. Continuing the previous problem, please write a program that automatically generate the number of subintervals \( n \) and calculate the Riemann integral given the user input of the interval boundary \( a \) and \( b \). You can build your program using the function you wrote for the previous problem. You can set a convergence tolerance as a named constant. You can also set a maximum iteration number to prevent from dead loop.

```fortran
program riemannIntegral
  implicit none
  real::a=0,b=0
  real,parameter::Delta=1e-6
  real,parameter::MaxIteration=1e6
  real::currR=0,prevR=0,diffR=1e10
  integer::n=10, step=10
  real::riemannSum

  write(*,*) “enter the boundaries a and b:”
  read(*,*) a,b
  currR = riemannSum(a,b,n)
  do while( diffR > Delta .and. n < MaxIteration )
    prevR = currR
    n = n + step
    currR = riemannSum(a,b,n)
    diffR = abs( currR - prevR )
  end do
  write(*,*) “The integration result is:”, currR
  write(*,*) “The difference between the results from ”, n, “ vs. ”, n-step, “ intervals is:”, diffR
end
```

```fortran
real::riemannSum
```