Final Exam (Close book, 1 A4 double-sided cheat sheet, 120 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 exam.

Signature_______________________________________________________________

Multiple Choice (2.5 points each)
Each problem has one correct answer. Clearly write the letter corresponding to the correct answer in the boxes on the right.

1. A Central Processing Unit (CPU) in a computer is composed of:
   A. Arithmetic Logic Unit
   B. Registers
   C. Control Unit
   D. All of the above
   D

2. Which computer hardware stores all the data declared in a Fortran program during its execution?
   A. Random Access Memory
   B. Hard-drive
   C. CPU registers
   D. Monitor
   A

3. What is the smallest data unit stored in a computer?
   A. word
   B. bit
   C. byte
   D. double word
   B

4. Which of the following declarations consumes the most memory space during run-time execution?
   A. Character::valid=.false.
   B. Real::val=0.0
   C. Integer,dimension(2)::a=(/0, 0/)
   D. Real,dimension(2)::b=(/0.0, 0.0/)
   D
5. Which type of Fortran statements triggers CPU operations?
   A. Comment statements
   B. User-defined function declaration statements
   C. Executable statements
   D. Variable declaration statements
   - C

6. When integer data type is used in arithmetic and assignment operations, which of the following issues might trigger run-time or logic bugs?
   A. Integer truncation
   B. Division by zero
   C. Integer overflow
   D. All of the above
   - D

7. In order to make an edited Fortran source code of a computer program executable, at least the source code needs to be
   A. compiled
   B. re-edited
   C. debugged
   D. run
   - A

8. A declared variable holds undetermined value until it is assigned. Which of the following Fortran statement can determine an uninitialized real typed variable named `foo` during run-time?
   A. `write(*,*) “foo=”, foo`
   B. `foo = foo * 2`
   C. `result = foo`
   D. `read(*,*) foo`
   - D

9. Which of the following data holder can be initialized?
   A. Allocatable array
   B. Argument with attribute of intent(in)
   C. Local variables
   D. All of the above
   - C

10. Other than the monitor, where else can a Fortran write statement write?
    A. A character typed variable
    B. A constant
    C. An integer array
    D. Keyboard
    - A

11. Which Fortran logical expression is Illegal?
    A. `3 > 2.0`
    B. `(5*3) .or. 1.0`
    C. `1 /= 0`
    D. `(5>3) .or. ( 2 == 6 )`
    - B

12. Which Fortran logical expression is .true.?
    A. `7 == 6`
    B. `.not.( ‘A’ == ‘a’ )`
    C. `7/6 > 1`
    D. `2**3 < 8`
    - B
13. Which Fortran branching clause is appropriate for the implementation of the following equation

\[ \text{sinc}(x) = \begin{cases} 
1, & \text{if } x = 0 \\
\frac{\sin(x)}{x}, & \text{if } x \neq 0 
\end{cases} \]

A. if ... end if
B. if ... else ... end if
C. if ... else ... end
D. if ... else if ... else ... end if

B

14. Assume all the variables are declared as integer, what is the value of the variable \( t \) after the following loop construct?

\[
t = 0 \\
i = 1 \\
\text{do while (} i < 4 \, \text{)} \\
\quad t = t + i*i \\
\quad i = i + 2 \\
\end{do}
\]

A. 9
B. 10
C. 12
D. 14

B

15. Assume all the variables are declared as integer, what is the value of the variable \( t \) after the following Fortran statement?

\[
t = 0 \\
\text{do i=1,3} \\
\quad t = t + i \\
\text{end do}
\]

A. 3
B. 4
C. 6
D. 7

C

16. Which Fortran statement is valid given the following array declaration:

\[
\text{real, dimension(4)::arr=0}
\]

A. \( \text{arr(1)=arr(2)+arr(3)+arr(4)} \)
B. \( \text{arr(0)=1.0} \)
C. \( \text{arr='test'} \)
D. \( \text{arr(2)=true}. \)

A

17. Which statement does NOT apply to calling user-defined functions in Fortran?

A. User-defined function needs to be declared
B. Function argument list needs to be passed during function call
C. User-defined function does not return anything
D. Function calls make code re-usable

C
18. Which statement does NOT apply to Fortran subroutine?
   A. Subroutine can only be called in the Call statement
   B. Subroutine can have multiple output arguments
   C. Subroutine needs no declaration
   D. Subroutine cannot call function

19. Which Fortran statement declares a 2-dimensional array of 12 elements?
   A. integer,dimension(12)::a1
   B. real,dimension(3,4)::a2
   C. integer,dimension(:), allocatable::a3
   D. real,dimension(10,2)::a4

20. If the memory for an array is allocated during run-time, which kind of Fortran statements needs to be used to avoid memory leak?
   A. Deallocate statement
   B. Assignment statement
   C. Termination statement
   D. “Implicit none” statement
21. (10 points) Read the following partial Fortran code:

```fortran
integer, dimension(6)::arr=(/10, -1, 3, 2, 9, 0/)
integer::i=0, j=0, t=0
! first loop
do i=1,6
   if ( arr(i) < 0 ) then
      arr(i) = -arr(i)
   end if
end do
! second loop
  t = 0
  do i=1,6
     t = t + arr(i)/2
  end do
```

1) Write the value of array variable `arr` right after the first loop and before the second loop:

- arr(1): 10
- arr(2): 1
- arr(3): 3
- arr(4): 2
- arr(5): 9
- arr(6): 0

2) Write the value of the integer variable `t` after the second loop:

11

arr(i)/2 is operated on integers for both the nominator and the denominator. Thus integer truncation operation is applied to arr(i)/2.

<table>
<thead>
<tr>
<th>Iter#</th>
<th>arr(i)/2</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10/2=5</td>
<td>0+5=5</td>
</tr>
<tr>
<td>2</td>
<td>1/2=0</td>
<td>5+0=5</td>
</tr>
<tr>
<td>3</td>
<td>3/2=1</td>
<td>5+1=6</td>
</tr>
<tr>
<td>4</td>
<td>2/2=1</td>
<td>6+1=7</td>
</tr>
<tr>
<td>5</td>
<td>9/2=4</td>
<td>7+4=11</td>
</tr>
<tr>
<td>6</td>
<td>0/2=0</td>
<td>11+0=11</td>
</tr>
</tbody>
</table>
22. (10 points) Read the following Fortran program partial code

```fortran
integer::i=0,j=0
real::tmp=0.0
real, dimension(3,3) :: a = 0
real, dimension(9) :: b = (/1,3,0,2,4,0,0,0,5/)
! first nested loop
do i = 1,3
    do j = 1,3
        a(i,j) = b(i*3+j)
    end do
end do
! second nested loop
do i = 1,3
    do j = i+1,3
        tmp = a(i,j)
        a(i,j) = a(j,i)
        a(j,i) = tmp
    end do
end do
```

a) Write the 2-dimensional array variable a in a matrix after the first nested loop and before the second nested loop:

\[
\begin{bmatrix}
2. & 4. & 0. \\
0. & 0. & 5. \\
x & y & z
\end{bmatrix}, \text{ where } x, y, \text{ and } z \text{ represent any real values.}
\]

b) How many iterations are there in the second nested loop?

3

c) Write the 2-dimensional array variable a in a matrix after the second nested loop:

\[
\begin{bmatrix}
2. & 0. & x \\
4. & 0. & y \\
0. & 5. & z
\end{bmatrix}
\]
23. (10 points) The following Fortran function calculate the factorial of a non-negative integer using the following equation.

\[ n! = \begin{cases} 
1, & \text{if } n = 0 \\
\prod_{k=1}^{n} k & \text{if } n > 0
\end{cases} \]

There are two bugs in the implementation. Fix the errors directly in the code.

```fortran
integer function factorial( n )
    implicit none
    integer, intent(in)::n
    integer::r = 0, k = 0
    if ( n == 0 ) then
        r = 1
    else if ( n > 0 ) then
        r = 1
        do k = 1,n
            r = r * k
        end do
    end if
    factorial = r
end function
```

Bug1: == instead of = for relational logical expression
Bug2: r needs to be set to 1 before the accumulative multiply loop starts, otherwise, r stays zero after the loop.
Bug3: integer k used in the loop needs to be declared

Fixing any two out of the 3 bugs will gain full credit.
24. (20 points) Given a 2x2 matrix of \( x \)

\[
  x = \begin{bmatrix}
  x_{11} & x_{12} \\
  x_{21} & x_{22}
  \end{bmatrix}
\]

its inverse \( y \) can be calculated as follows

\[
y = x^{-1} = \begin{bmatrix}
  y_{11} & y_{12} \\
  y_{21} & y_{22}
  \end{bmatrix} = \begin{bmatrix}
  x_{22} & -x_{12} \\
  -\frac{x_{11}x_{22} - x_{12}x_{21}}{x_{22}} & \frac{x_{11}x_{22} - x_{12}x_{21}}{x_{22}}
  \end{bmatrix}
\]

Implement 2x2 matrix inversion in a Fortran subroutine given an input array and an output array as its arguments. Both arrays are in the shape of 2x2 and are in real type. You can assume the value of \( x_{11}x_{22} - x_{12}x_{21} \) is non-zero in the subroutine.

```fortran
subroutine inv( x, y )
  implicit none
  real, dimension(2,2),intent(in)::x
  real, dimension(2,2),intent(out)::y
  real::scale=0
  scale = 1.0 / (x(1,1)*x(2,2) - x(1,2)*x(2,1))
  y(1,1) = x(2,2) * scale
  y(1,2) = -x(1,2) * scale
  y(2,1) = -x(2,1) * scale
  y(2,2) = x(1,1) * scale
end subroutine
```
Part IV. Optional problem (20 points)

25. Write a Fortran function to implement the following equation:

\[ y(x) = \sum_{n=1}^{N} \frac{(-1)^{n-1} (x-1)^n}{n} \]

The input of the function is a real typed argument of \( x \), and an integer typed argument of \( N \). The output of the function is a real typed value of \( y(x) \).

```fortran
real function logApprox( x, N )
  implicit none
  real, intent(in)::x
  integer,intent(in)::N

  integer::i=0
  real::result=0

  do i=1,N
    result = result + (-1)**(i-1) * (x-1)**i / i
  end do

  logApprox = result
end function
```