

Name: Key

Grade: _____/100

1. For the following errors, describe how you would correct the problem identified.

1. (3 points)

```
[lejohnso@isengard lejohnso]$ g95 junk.f95
x = 3*y + z**2
```

1

Error: Symbol 'x' at (1) has no IMPLICIT type

declare x
REAL:: x

2. (3 points)

```
[lejohnso@isengard lejohnso]$ g95 junk.f95
In file junk.f95:7
```

```
z = x**2 + 2x - y
```

1

Error: Unclassifiable statement at (1)

change to

$$z = x**2 + z * x - y$$

3. (3 points)

```
[lejohnso@isengard lejohnso]$ g95 junk.f95
In file junk.f95:10
```

```
WRITE(*,*) "X = ", x, " Y = ", y, "Z = ", z
```

1

Error: Syntax error in WRITE statement at (1)

add "

~~WRITE(*,*)~~ "X = ", x, " Y = ", y, "Z = ", z

4. What is the value of *result* after the following statements are executed? (6 points)

```
REAL:: a=2.4, b=1.2, c=5., result
```

```
result = a / b + b * c**2
```

24

2 + 30

result = 32.0

3. (8 points) What values will be output by the following program?

```

PROGRAM SAMPLE
INTEGER:: i1, i2, i3, i4
REAL:: a1 = 2.4, a2
  i1 = a1
  i2 = -a1 * i1
  i3 = -a1 * i1 - 0.5
  i4 = a1 * i1 / 2.0
  WRITE(*,*) i1, i2, i3, i4, a1, a2
END PROGRAM

```

$i1 = 2$
 $i2 = -4$
 $i3 = -5$
 $i4 = 2$
 $a1 = 2.4$
 $a2 = ??$

2 -4 -5 2 2.4 junk

4. Determine the value, .TRUE. or .FALSE. for each of following logical expressions, assuming that the value of the variable count is 0 and the value of the variable limit is 10. Give your answer as the values .TRUE. or .FALSE. (10 Points)

- a) (count == 0) .AND. (limit < 20)
- b) limit > 20 .OR. count < 5
- c) .NOT. (count == 12)
- d) count >= 0 .AND. count <= 10 .AND. limit < 5
- e) count >= 0 .AND. count <= 10 .OR. limit < 5

.TRUE.

~~.TRUE.~~

.TRUE.

.FALSE.

.TRUE.

5. Rewrite the following cascading IF THEN ELSE statements into a SELECT CASE statement. Assume that grade is a Integer. (18 points)

```

IF (grade >= 90) THEN
  WRITE(*,*) "You got an A!"
ELSE IF (grade >= 80) THEN
  WRITE(*,*) "You got a B."
ELSE IF (grade >= 70) THEN
  WRITE(*,*) "You got a C."
ELSE IF (grade >= 60) THEN
  WRITE(*,*) "you got a D."
ELSE
  WRITE(*,*) "Consider changing majors."
END IF

```

SELECT CASE (grade)

CASE (90:100)

WRITE(*,*) "you got an A"

CASE (80:89)

WRITE(*,*) "you got a B"

CASE (70:79)

WRITE(*,*) "C"

CASE (60:69)

WRITE(*,*) "D"

CASE DEFAULT

WRITE(*,*) "change"

END SELECT

6. Write a piece of code (only a piece of code not a complete program), that prints to the screen values from 10 to -10 counting down by 2 each time. (8 Points) Use a counting do loop.

```

INTEGER :: i
DO i = 10 10, -10, -2
  WRITE(*,*) i
END DO

```

7. What is the output of the following code? (8 Points)

```

INTEGER :: M, Last_M = 3, N, Last_N = 2, product
DO M = 1, Last_M
  DO N = 1, Last_N
    product = M*N
    WRITE(*,*) M, " ", N, " ", product
  END DO
END DO

```

```

1 1 1
1 2 2
2 1 2
2 2 4
3 1 3
3 2 6

```

8. Determine which of the following are valid FORTRAN labels (variable names). Assume that you are using the Fortran 95 standard. If they are not valid explain why. (8 Points)

- a. variable_x *valid*
- b. 4LifelsGood *not valid - starts with #*
- c. what? *not valid - has a?*
- d. end *not valid - Reserved word*
- e. begin&end *not valid - has &*
- f. where UR *not valid - has space*
- g. are_you_sick_of_political_advertisements_yet *not valid - more than 31 characters*

9. Using Fortran order of operations, what is the results of the following mathematical operations. (10 Points)

a) $2 * 5 + 3 * 4$ $10 + 12 = 22$

b) $13 / 4 + 3$ 6

c) $2 ** 2 * 3.14$ 12.56

d) $4.5 / 3.0 - 1.0$ 0.5

e) $2 ** 3 ** 2$ 512

10. Show the results of the following binary to base 10 conversions or base 10 to binary conversions. (8 points)

a. 101101_2 $32 + 8 + 4 + 1 = 45$

b. 01011000_2 $64 + 16 + 8 = 88$

c. 109_{10} $64 + 32 + 8 + 4 + 1 = 01101101$

d. 61_{10} $32 + 16 + 8 + 4 + 1 = 00111101$

11. Write a program that asks the user for a positive integer value. The program should use a do loop to get the sum of all the integers from 1 up to the number entered. For example, if the user enters 50, the loop will find the sum of 1, 2, 3, 4, ..., 49, 50. Input Validation: Do not accept a negative starting number. (16 Points)

```
PROGRAM P11d011
IMPLICIT NONE
INTEGER :: i, n, sum

sum = 0
WRITE(*,*) "enter a positive integer"
READ(*,*) n
IF (n <= 0) THEN
  DO i = 1, n
    sum = sum + i
  END DO
  WRITE(*,*) "The sum is", sum
ELSE
  WRITE(*,*) "user input error"
ENDIF
END PROGRAM
```