

CS 1014: LOOPS

Study chapter 6 thoroughly from the text and chapters 6 and 7 from the course notes to learn about loops. You must know the syntax and the use of the *do while* statement, the *do* statement, the *cycle* statement, and the *exit* statement.

Practice Problems:

1. **Finding Errors:** Correct each of the following code segments if there is any error. You may try writing a small program for each of them. Be careful about infinite loops!!!

a. INTEGER :: num = 1
 DO WHILE (num = 1)
 WRITE (*,*) num, (num + 1), (num + 2)
 num = 0
 END DO

b. INTEGER :: num = 3
 DO WHILE (num /= 0)
 WRITE (*,*) num
 num = 0
 END DO

c. INTEGER :: count = 0, total = 0
 DO WHILE (count < 7)
 total = total + count
 count = count - 1
 END DO

2. **Short Questions:** Answer the following short questions on loops. You may type, compile, and run code segments to check your answers. Do not forget to include proper declarations in each of your test program.

- a. How many times does the following loop display Hello?

```
INTEGER :: count = 3
LOGICAL :: doit = .TRUE.
DO WHILE (doit)
    count = count - 1
    WRITE (*,*) "Hello"
    if (count == 0) doit = .FALSE.
END DO
```

- b. What output does the following code segment produce?

```
INTEGER :: i, sum = 0
DO i = 1, 20
    IF (MOD(i, 2) == 1) CYCLE
    sum = sum + i
END DO
WRITE (*,*) sum
```

- c. What output does the following code segment produce if the input from the keyboard is 29? What will be the output if the input is 35 instead?

```
INTEGER :: i, n
READ (*,*) n
DO i = 2, n
    IF (MOD(n, i) == 0) EXIT
END DO
IF (i < n)
```

```

        WRITE (*,*) n, " is divisible by " , i
    ELSE
        WRITE (*,*) n, " is a prime "
    END IF

```

- d. For each *do* statement find the number the loop body executes and the value of the loop counter when the program exits the loop.
 - i) DO i = 1, 100, 3
 - ii) DO i = 100, 1, -2
 - iii) DO i = -5, 104, 5
 - iv) DO i = 25, -10, -2
3. **Coding:** It is important to learn how to code loops in Fortran. Here are some simple coding problems that you should try to code in Fortran.
 - a. Write Fortran statements to determine the sum of odd integers from 3 to 999.
 - b. Write Fortran statements to display even integers between 100 and 200 inclusive.
 - c. Write Fortran statements to determine the sum of squares from 1 to 20.
 - d. Write Fortran statements to add all the numbers in a file.
 - e. Write Fortran statements to read numbers repeatedly from the keyboard until a negative number is entered, and then calculate the average of all non-negative numbers.
 - f. Write a program that prompts the user to enter a number n, then displays all squares between 1 and n. For example, if the user enters 20, the program should display the following:


```

                        1
                        4
                        9
                        16
                    
```
 - g. Suppose that you deposit \$1.00 into an account this month, \$2.00 into the account the second month, \$4.00 into the third month, and so on, doubling your deposit each month. Write a program that determines and displays the first month that your deposit exceeds \$1,000,000.00.
 - h. Write a program that displays a table of integers from 1 to 20 with their squares and cubes.
 - i. Write a program that displays a conversion table from Fahrenheit to Celcius between 25 degrees Fahrenheit through 105 degrees Fahrenheit in steps of 5 degrees. The conversion formula is

$$\text{Celcius} = 5 * (\text{Fahrenheit} - 32) / 9$$

Final Note: The next step of learning "loops" is to understand and try the example programs posted on the web and the example programs found in the text book. For those who do not have any prior programming experience, I would suggest you try Programming Problems 6.3 and 6.12 (pages 196, 198) from the textbook for more exercise.