

## Practice Exercises on Loops

**Problem 1:** How many times will the *for loop* in the following code run and what will the output be? Work this out by hand and just use MATLAB to check your answer.

```
x = 2; sum = 1;
for k = 1:5
    sum = 1 + 1/x*sum;
end
disp('sum =');disp(sum)
```

**Problem 2:** What will the following code produce? Work this out by hand then check result in MATLAB

```
% Geometric Series
x = 4; N = 5;
series = 0;
for m = 1:N
    series = series + x^m;
end
fprintf('The sum for the geometric series with x = %0.5f and %d
terms is: %0.5f \n',x,N,series)
```

**Problem 3:** How many times will the *while loop* in the following code run and what will the output be? Again, work this out by hand and just use MATLAB to check your answer.

```
sum = 0;
while sum <=10
    sum = sum + 3;
end
disp('sum =');disp(sum)
```

**Problem 4:** The wind chill factor (WCF) describes how cold it “feels” for a given temperature T, in Fahrenheit, and a given wind speed V (in miles per hour). The equation for wind chill factor is:

$$\text{WCF} = 35.7 + 0.6T - 35.7(V^{0.16}) + 0.43T(V^{0.16})$$

Write a script that has three inputs: temp in Fahrenheit, minimum wind speed, and maximum wind speed. Use a *for loop* to compute and display the wind chill factor (WCF) using the given temperature over the wind speed range in increments of 5 miles per hour. For example, suppose the user inputs a temperature of 20°, a minimum wind speed of 5 mph., and a maximum wind speed of 20 mph. The output of the program should look like this (that is, use fprintf to insert your calculated values into the text):

```
For a temperature of 20 degrees F
The wind chill factor for a wind speed of 5 m.p.h. is: 12.6 degrees F
```

The wind chill factor for a wind speed of 10 m.p.h. is: 8.5 degrees F  
The wind chill factor for a wind speed of 15 m.p.h. is: 5.9 degrees F  
The wind chill factor for a wind speed of 20 m.p.h. is: 3.9 degrees F

**Problem 5:** Consider the following simple savings plan. On Day 1 you put aside one penny. On Day 2, you put aside two pennies. On Day 3 you put aside three pennies. You continue this simple savings plan for several years.

How much money do you think you will have in 30 years? Make a guess: \_\_\_\_\_

Write a script that will compute the amount of money accumulated for a specified number of years. The input to your program will be number of years. The outputs of your program will be the amount of money saved (in dollars, not pennies) and the amount of money contributed on the very last day of your savings plan (in dollars, not pennies).

*Note: Don't worry about leap years – just assume 365 days per year.*

Use your function to complete the following table:

Years for Savings Plan	Accumulated Savings	Contribution on Final Day
20		
30		
40		

**Problem 6:** Write a MATLAB script that will allow a user to play the dice game: Under and Over Seven. The rules are pretty simple:

1. The game is played with two dice
  2. The player specifies how much money he/she is betting on the roll
  3. The player bets whether he/she will roll Under 7, Over 7, or Exactly 7
  4. The player specifies how much money he/she is betting on the roll
  5. The player rolls the dice
  6. If the player bets incorrectly, he/she loses the amount of money in the bet
  7. If the player bets correctly, the payoff is 1:1 for Under 7 and for Over 7 (that is, if the bet was \$1 then the player wins \$1) and is 4:1 for Exactly 7 (that is, if the bet was \$1 then the player wins \$4 for betting on and successfully rolling a 7).
- Your program should begin by asking the player the total amount of money that he/she has to play this game. Throughout the game, your function will keep track of how much money the player has left based on wins and losses.
  - The player should then be asked to place his/her bet (Over 7, Under 7, Exactly 7) and the amount of money for the bet. If the player tries to bet more than he/she has, prompt for a new bet amount.
  - Roll the two dice and display the results (the number on each dice and win or lose). Use the *randi* function to do this.

- Calculate and display to the user his/her new balance (total amount of money he/she has now).
- If the balance is greater than zero, ask the user if he/she would like to play again. If the user says answers yes, prompt for a bet and a betting amount and run through the cycle again.
- The game (program) should end when the user has no money left to bet or the user decides to quit.