

## MATLAB Marina: Iteration, while loops

1. Create a MATLAB program consisting of the MATLAB code of Figure 1. Run the program and verify that the values in the array `numbers` are displayed. Determine the size and value(s) of the variable `k` after the loop has been executed for the program.

---

```
clear; clc;
numbers = 1:0.5:5;
k = 1;
while k <= length(numbers)
    val = numbers(k);
    fprintf('%0.1f ', val);
    k = k + 1;
end
fprintf('\n');
```

---

Figure 1. Display Numbers Code Segment

2. Write a MATLAB program that will display the numbers from 0 to 45 with a step of 5. The program should display: 0, 5, 10, 15, 20, etc.. Use iteration (while loop) to solve this problem. Run your program and verify that it operates correctly.
3. Modify the program of Exercise 2 so that the numbers are displayed in reverse order: 45, 40, 35, ..., 5, 0.. Use iteration (while loop) to solve this problem. Run your program and verify that it operates correctly.
4. For operations like those in Exercises 1-3, is a for loop or while loop a better choice? Explain why.
5. The MATLAB program of Figure 2 ensures that a number read in from the user will be greater than zero. Modify the program of Figure 2 to ensure a number between 1 and 100 is read in. Test the program for the inputs -5, 1, 50, 100, and 105 to verify program operation.

---

```
clear, clc,
% read in number
number = input('Enter number greater than zero: ');
% ensure number is greater than 0
while (number <= 0)
    number = input('Enter number greater than zero: ');
end
```

---

Figure 2. Program Segment Ensuring Data is Valid

6. Complete the MATLAB program of Figure 3 that will sum positive numbers (numbers greater than or equal to zero) read in from the user until a negative number is entered. Once the negative value is entered, the sum of the all the numbers previously entered

should be displayed. This is a running sum problem, but a while-loop should be used since the number of values is not known beforehand. The sentinel value (the negative value that indicates the end) should not be added to the sum.

---

```
clear, clc,
sumNumbers = 0;
% read in first number
number = input('Enter number (negative to end): ');

% while number is not negative update sum

    % update sum of numbers

    % read in next number
```

---

Figure 3. Program to Sum Positive Numbers

7. Complete the MATLAB program of Figure 4 that will build (populate) a column vector with values read in from the user until the value -999 is entered. Once the sentinel value -999 is entered, the column vector should be displayed. Assume all the values other than the sentinel will be between -500 and 500. This is a running array concatenation problem, and a while-loop should be used since the number of values to populate the array with is not known beforehand. The sentinel value should not be part of the resulting array.

---

```
clear, clc,
% empty initial column vector
colVector = [];
% read in first number
number = input('Enter number (-999 to end): ');

% while number is not -99 negative build column vector

    % concatenate new number to existing vector

    % read in next number
```

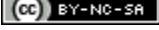
---

Figure 4. Program to Build Column Vector

8. Write a MATLAB program that will approximate the infinite series  $\sum_{n=0}^{\infty} \frac{(-1)^n}{2^n}$ . The approximation should be accurate to five decimal places. An alternating series that converges has an error bound equal to the magnitude of the next term in the series, for example if the series was approximated using  $N = 3$ ,  $\sum_{n=0}^3 \frac{(-1)^n}{2^n}$ , the error will be less than or

equal to  $\frac{(-1)^4}{2^4} = \frac{1}{16} = 0.0625$ . This problem should be solved using a while loop that terminates when the error bound is smaller than the desired accuracy.

Last modified Wednesday, March 30, 2022

 [MATLAB Marina](#) is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.