

Task 1

The value of $\cos(x)$ can be approximated using a Maclaurin series

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

which can be expressed more compactly as

$$\sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{(k-1)*2}}{((k-1)*2)!}$$

(recall that the symbol ! stands for factorial).

Use a midpoint break loop to determine how many terms must be included in the summation, in order to find the correct value of $\cos(2)$ within an error of .001. Limit the number of iterations to a maximum of 10.

Task 2

Use for loop to sum the elements in the array

X = [1 23, 43 72 87 56 99 33]

Compare with sum function

Repeat with while loop

Task 3

A Fibonacci sequence is composed of elements created by adding the two previous elements. The simplest Fibonacci sequence starts with 1, 1 and proceeds as follows:

1, 1, 2, 3, 5, 8, 13, ...

However, a Fibonacci sequence can be created with any two starting numbers. Fibonacci sequences appear regularly in nature. For example, the shell of the chambered nautilus (Figure P9.12) grows in accordance with a Fibonacci sequence.

Prompt the user to enter the first two numbers in a Fibonacci sequence and the total number of elements requested for the sequence. Find the sequence and store it in an array by using a `for` loop. Now plot your results on a polar graph. Use the element number for the angle and the value of the element in the sequence for the radius.

Repeat using while loop



Figure P9.12

Task 4

1. Create a table that converts inches to feet.
2. Consider the following matrix of values:

$$x = [45, 23, 17, 34, 85, 33]$$

How many values are greater than 30? (Use a counter.)

3. Repeat Exercise 2, this time using the `find` command.
4. Use a `for` loop to sum the elements of the matrix in Problem 2. Check your results with the `sum` function. (Use the `help` feature if you don't know or remember how to use `sum`.)

Repeat 1 – 4 in task 4 using while loop

Task 5

Assume that the following are bar codes, and next to them are prices related to items

barCodes = [1 2 3 4 5 6 7 8 9 10]'

prices = [10 12 23 14 55 16 37 84 19 99]'

list = [barCodes prices]

use while loop and prices to determine the amount of money needed to be paid as long as there are items on the tray, and if -1 is entered the bill is printed

HW tasks

Task 1

Write a script to solve this problem. Assume you have a vector named D. Using iteration (for and/or while) and conditionals (if and/or switch), separate vector D into four vectors posEven, negEven, posOdd and negOdd.

- posEven contains all of the positive even numbers in D.
- negEven contains all of the negative even numbers in D.
- posOdd contains all of the positive odd numbers in D.
- negOdd contains all of the negative odd numbers in D.

For example:

```
if D = [-4,-3,-2,-1,0,1,2,3,4],  
posEven=[2,4], negEven=[-4,-2],  
posOdd=[1,3] and negOdd=[-3,-1]
```

Task 2

You have a friend who has too many clothes to store in her/his tiny wardrobe provided by Georgia Tech. Being a good friend, you offer to help to decide whether each piece of clothing is worth saving. You decide to write a script in MATLAB that will compute the value of each piece of clothing. A piece of clothing has five attributes that can be used to determine its value. The attributes are:

condition, color, price, number of matches, and comfort.

Each attribute will be rated on a scale of 1-5. Write a script called `clothes` in MATLAB that will ask the user for the ratings for each attribute and store the result in a vector. The order of attributes in the vector are as follows:

```
[condition color price matches comfort]
```

The script should compute a value between 0 and 100; 100 represents a good piece of clothing while 0 represents a bad piece of clothing.

The points that should be given for each attribute are shown below:

Condition: 1=>0; 2=>5; 3=>10; 4=>15; 5=>20

Color: 1 => blue => 12;
2 => red (UGA Colors) => 2;
3 => pink => 15;
4 => yellow (GT Colors) => 20;
5 => white => 12

Price: 1 => 8, 2-3 => 16, 4-5 => 20

Matches: 1-2 => 8, 3-5 => 19

Comfort: 1 => 6, 2-3 => 13, 4-5 => 18

Note: If a number other than 1-5 is assigned for one of the attributes, no points should be given.

Task 3

You must use either `for` or `while` to solve the following problems.

- a. Iterate through a vector, `A`, using a `for` loop and create a new vector, `B`, containing logical values. The new vector should contain `true` for positive values and `false` for all other values.

For example, if

`A = [-300 2 5 -63 4 0 -46]` the result should be

`B = [false true true false true true false]`

- b. Iterate through the vector, `A`, using a `while` loop and return a new vector, `B`, containing `true` for positive values and `false` for all other values.

- c. Iterate through a logical array, `N`, using a `for` loop and return a new vector, `M`, containing the value 2 wherever an element of `N` is true and the value -1 (not a logical value) wherever `N` is false. For example, if

`N = [true false false true true false true]` the result should be `M = [2 -1 -1 2 2 -1 2]`

- d. Iterate through an array, `Z`, using a `while` loop. Replace every element with the number 3 until you reach a number larger than 50. Leave the rest unchanged. For example, if

`Z = [4 3 2 5 7 9 0 64 34 43]`, after running your script,

`Z = [3 3 3 3 3 3 3 3 34 43]`

Task 4

What are the problems you are facing learning this material and course?

Task 5

Modify Task 5 from tutorial, so that `barCode` is entered, but total money is calculated.