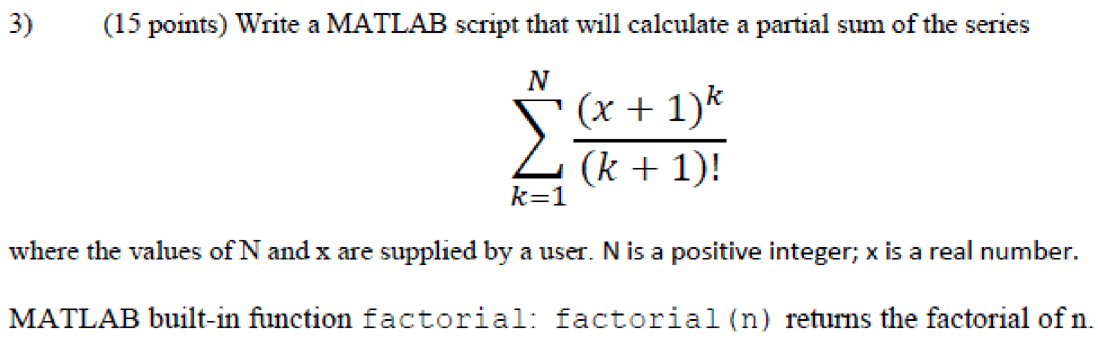
The following are (hacked/badly) problems from prior year exams or exam study problems I came up with. These were originally intended for MATLAB, but I've tried in most cases to update the wording to relate to Python. If you see "script", assume that equals "Python program". The conceptual parts of the questions remain valid, even if some of the specifics are a bit different. Use what's helpful from these, and I'll try and add some additional problems as I can.
-CS

Write a (program) that will ask a user to input a list of real numbers and then find and output the smallest and largest values. You may assume that the user input is valid (that is, if the user inputs a list, the list contains only real numbers – no need to check). If the user inputs an empty list, then your program should output an error message. Make sure to implement a loop in your script, that is, you cannot use the built-in functions such as `min`, `max`, `sort`, etc.



3) (15 points) Write a MATLAB script that will calculate a partial sum of the series

$$\sum_{k=1}^{N} \frac{(x+1)^k}{(k+1)!}$$

where the values of N and x are supplied by a user. N is a positive integer; x is a real number.

MATLAB built-in function `factorial`: `factorial(n)` returns the factorial of n.

Perfect numbers are numbers that are the sum of their factors. For example 6 is a perfect number, as its factors are 1, 2, and 3, and 1+2+3 = 6.

Abundant numbers are numbers that are less than the sum of their factors. For example, 12 is an abundant number, as its factors are 1, 2, 3, 4, and 6 and 1+2+3+4+6=16 which is greater than 12.

Deficient numbers are numbers that are greater than the sum of their factors. For example 10 is deficient, as its factors are 1,2 and 5, and 1+2+5=8 which is less than 10.

a. Write a (program) that determines if a number is perfect, abundant, or deficient. Do this by first finding the factors of the input number, then the sum of the factors, then compare the sum with the input number. Your program should take a number as input, and output a value 1 for perfect, 2 for abundant or 3 for deficient.

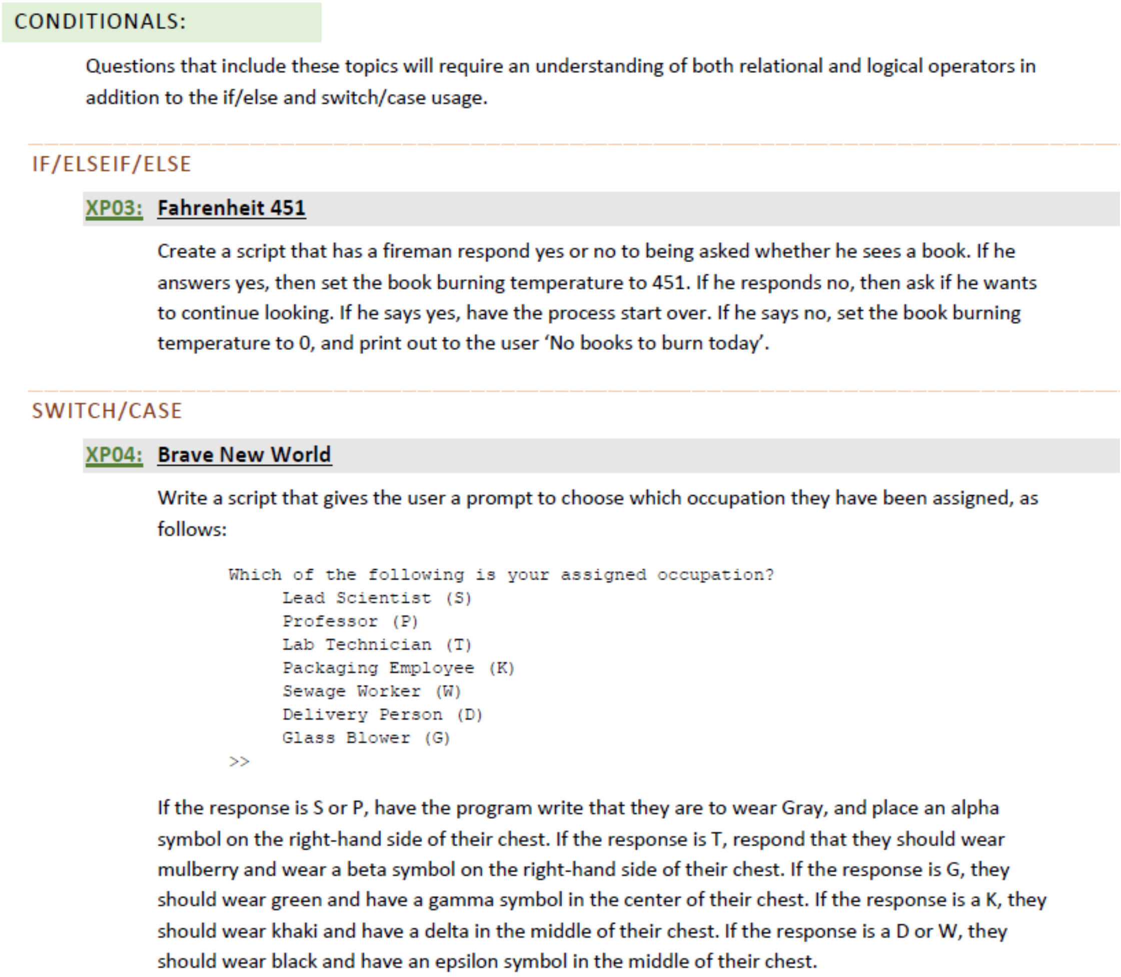
b. Have your program find how many perfect, abundant, and deficient numbers there are below 10,000. Print the result in a formatted manner.

Consider the following in a virtual dice game (using two random integers representing two dice):

I. If the sum of the two values is 7 or 11, write “Awesome, you win!”
II. If the sum of the two values is 2, 3 or 12, write “Too bad, try again!”
III. Otherwise, roll the two dice again.
    i. If the sum of the 2 new values equals the original sum, then write “Awesome, you win!”
    ii. If the sum of the 2 new values equals 7, write “Too bad, try again!”
    iii. Otherwise, roll the two dice again. Continue until either i or ii occur

Write a program to follow these game rules. Store the result of each roll in a list, and append each additional roll to this list. Display the final array to the user, along with the game result (“Awesome, you win!” or “Too bad, try again!”).

> *Hint:* `randint(1,6)` *will generate a random integer value between 1 and 6, and this requires you to import the “random” library at the start of your program*



CONDITIONALS:

Questions that include these topics will require an understanding of both relational and logical operators in addition to the if/else and switch/case usage.

IF/ELSEIF/ELSE

XP03: Fahrenheit 451

Create a script that has a fireman respond yes or no to being asked whether he sees a book. If he answers yes, then set the book burning temperature to 451. If he responds no, then ask if he wants to continue looking. If he says yes, have the process start over. If he says no, set the book burning temperature to 0, and print out to the user ‘No books to burn today’.

SWITCH/CASE

XP04: Brave New World

Write a script that gives the user a prompt to choose which occupation they have been assigned, as follows:

```
Which of the following is your assigned occupation?
     Lead Scientist (S)
     Professor (P)
     Lab Technician (T)
     Packaging Employee (K)
     Sewage Worker (W)
     Delivery Person (D)
     Glass Blower (G)
>>
```

If the response is S or P, have the program write that they are to wear Gray, and place an alpha symbol on the right-hand side of their chest. If the response is T, respond that they should wear mulberry and wear a beta symbol on the right-hand side of their chest. If the response is G, they should wear green and have a gamma symbol in the center of their chest. If the response is a K, they should wear khaki and have a delta in the middle of their chest. If the response is a D or W, they should wear black and have an epsilon symbol in the middle of their chest.

XP06: Winter

Write a script that generates a random integer between 0 and 100. Have the script tell you the number it chose, and also which numbers the generated number is divisible by. Have it count how many numbers it is divisible by, and then print out “Winter is coming!” the same number of times.

XP07: More Summer

Write a script that calculates the partial sum of the following series, where the value of *N* is supplied by the user:

$$\sum_{k=1}^{N} \frac{1}{2^k}$$

WHILE (INDEFINITE LOOPS)

XP08: Spring

Write a function that takes in a value you call a spring coefficient. Have the program generate a random number between a given min and max (make sure your spring coefficient is between the limits). Have it stop once the number it generates is greater than the spring coefficient and print out how many tries it took, and to say “pop goes the weasel!”

> *All around the mulberry bush*
> *The monkey chased the weasel*
> *The monkey thought ‘twas all in fun*
> *Pop! Goes the Weasel!*

XP09: Fall

Create a trace table, and determine how many iterations the following code will need (11).

```
skipLeaf = 2;  sweepLeaf = 0;  broomTime = 0;
while (sweepLeaf < 100)
   sweepLeaf  = sweepLeaf + broomTime * skipLeaf;
   broomTime = broomTime + 1;
   fprintf(‘sigh – grumble’)
end
fprintf (‘Mom, I’m done!\n’)
```

XP15: Was it a car or a cat I saw?

Write a function (`ispalindrome.m`) that takes a single input argument which is a string that does not contain spaces and returns a single output argument which is a logical value representing whether or not the input string is a palindrome.

A palindrome is a string, such as a word or phrase, which is the same forwards as backwards. For example, the string “Madam, I’m Adam” is a palindrome when punctuation, whitespace, and capitalization are ignored: madamimadam ←→madamimadam. Other examples of palindromes include: “taco cat”, ”radar”, “a toyota”,”never odd or even”, “a man a plan a canal panama”. When the input is a palindrome, the output should be the logical value true. When the input is not a palindrome, the output should be the logical value false.

You may assume that the input is a string and that it does not contain spaces or punctuation.