

ENGR 102 – Fall 2023
Exam 1 – MORE Practice Problems

During the exam calculators are NOT allowed, you won't need one anyway.

You may NOT use your phone. In addition, you may NOT use the web to search for additional information, your book, your notes, lectures on eCampus, or any form of electronic media.

Do not use `sympy`, `numpy`.

Disclaimer: You may (and will) see other types of problems/questions on Exam 1.

Code writing problems:

1. Write a Python program in the box below that takes in an integer between one hundred and one million, inclusive. You may assume the user always enters an integer. If the user enters a value outside the interval, print an error message. For any valid input, check the last two digits in the number: if both are even, print their sum; if both are odd, print their product. Otherwise, print 'One odd, one even!'

You may use the built-in `len()` function. Do NOT use loops, lists/tuples, or the `sort()` and `sorted()` functions, or any of the string class methods. Use four (4) spaces in place of one (1) tab. Don't forget to comment your code! Use good coding practices.

Example output using different input values:

```
Enter an integer between 100 and 1000000, inclusive: 15
Wrong input!
```

```
Enter an integer between 100 and 1000000, inclusive: 157
Both odd! Product = 35
```

```
Enter an integer between 100 and 1000000, inclusive: 3468
Both even! Sum = 14
```

```
Enter an integer between 100 and 1000000, inclusive: 12345
One odd, one even.
```

2. Write a Python program to take as input 5 birthdays from 5 users (1 each) and output them in chronological order. Dates should be entered with the month and day (not year) in the format "June 6" as a single input per user. You may format the output however you like (including using numbers for the month instead of words). This is a good problem to practice using dictionaries.

Exam 1 – MORE Practice Problems

3. Write a Python program to play a simplified version of the game hangman. Have User 1 input a secret word with a minimum length of 6. Then, take as input from User 2 one letter at a time until they guess a letter not contained in the secret word. At the end of the program, print out the number of guesses and the secret word.

Example:

- User 1 enters the secret word “programming”.
- User 2 guesses the letter “n”
- User 2 guesses the letter “a”
- User 2 guesses the letter “e”
- The program outputs the number of guesses (3) and the secret word (programming)

4. Write a Python program to take as input from the user a student’s UIN. If the UIN exists in the list `roster`, have your program output the first and last name of the student, their major, and their GPA. The list `roster` is a nested list and you may assume that it is available in the code (simply hard-code):

```
roster = [ sublist1, sublist2, etc.]
```

where each sublist is a list that contains the following information: UIN (string) and a list with first name (string), last name (string), major (string), and GPA (float). Here is an example of a sublist:

```
['123004567', ['Joe', 'Aggie', 'ENGE', 3.50]]
```

```
roster = [ ['123004567', ['Joe', 'Aggie', 'ENGE', 3.50]],  
          ['123004568', ['Jake', 'Green', 'OCEN', 3.75]],  
          ['123004569', ['Jill', 'Apple', 'ENGE', 3.80]] ]
```


5. Write a Python program to generate the following output exactly as shown using a loop.

```
a  
bb  
ccc  
dddd  
eeee
```

6. Write a Python program that will repeatedly ask a user to input a person's age. The program should continue to ask for input until a negative number is entered, indicating that the user is done inputting data. The program should determine the total number of people and the minimum and maximum ages entered. The results should be printed to the screen using the format shown below. Include the header and align the columns.

```
Number of people  Minimum age  Maximum age  
32                17            24
```

7. A schematic for converting phone letters to digits mapping is shown in the image below. Write a Python program that prompts the user to enter a 10-character phone number in this format XXX-XXXXXXX. Your program should replace the last seven alphabetic characters by their equivalent digits and display the entered phone number in this format XXX-XXX-XXXX. For example, if the user enters 800-GOFEDEX, your program output would be '800-GOFEDEX is equivalent to 800-463-3339'. You may assume that the last seven characters are alphabetic characters from A – Z.

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
*	0 +	#
+ 		

8. Write a Python code that can enter a value of n (n is a positive integer) and then calculate the sum of $n + nn + nnn$. For example, if $n = 12$, the sum is $12 + 1212 + 121212 = 122436$; if $n = 1$, the sum is $1 + 11 + 111 = 123$; if $n = 345$, the sum is $345 + 345345 + 345345345 = 345691035$.
9. For each integer n between 2 and 10 (inclusive), print numbers between n and $n*10$ that are multiples of n . You must use nested loops. Your output should look like this:

Integer Multiples

2 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
3 3, 6, 9, 12, 15, 18, 21, 24, 27, 30
4 4, 8, 12, 16, 20, 24, 28, 32, 36, 40
Etc.

10. Write a Python code that will ask the user to input a word (sentence) and then will reverse it using a for loop. Output the reversed sentence. For example, input 'howdy all!', output '!lla ydwoh'. Hint: recall that strings are not mutable, but you may build a new string, right?
11. Write a Python code that will ask the user to input a non-negative integer and then will add all the digits in the number. Output the sum.
12. Write a Python code that will ask the user to input a positive integer that contains at least 10 digits. Then ask the user to input a digit. Your code should remove that digit from the initial number and print the result. For example, the user inputs 3479734103487314 and 3. Your code outputs 479741048714.
13. Write a Python code that will ask the user to input 10 items that are sold in a school cafeteria (name, cost). Then ask the user to input the amount of money that the user has. The user wants to buy only one item. Print all items that the user can afford to buy.

14. Fibonacci numbers https://en.wikipedia.org/wiki/Fibonacci_number

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

Write a Python program that will ask the user to input a positive integer and then will print all members of the Fibonacci sequence up to that number (inclusive).

Write a program that will print the first n members of the sequence, where n is provided by the user.

15. Write a program that uses a loop (or nested loops) to create and print a nested list (m sub-lists with n elements each) with the following values:

[[0, 1, 2, 3], [1, 2, 3, 4], [2, 3, 4, 5]]

Hint: think of this list as of a 2D array or matrix, $m \times n$:

0	1	2	3
1	2	3	4
2	3	4	5

Note that each value is the sum of the row and column indices (if indices start at zero).

The number of rows and columns should be inputted by the user.

16. Write a program that will create a list of numeric values (user input) and then output the second largest value. You may assume that all values are unique. Think of two versions: simple – you may use the built-in function `max()`; a bit harder – you cannot use `max()`, `sort()`, etc. (that is, need a loop and one or two extra variables).

17. Write a program that will ask the user to input words until the user inputs 'stop' or 'Stop' or 'STOP' or 'StOp', etc. Assume that all words start with different letters. The program should then print all words inputted by the user and then print one word that should be printed first if you had to print the words in alphabetical order. The program should also count the number of words inputted by the user. You may NOT use any containers to store the words. "No containers" means "need to do processing in a loop."

Example: the user inputs 'dog', then 'howdy', then 'cat', then 'five', then 'red', then 'stop'. Your program prints the inputted words 'dog, howdy, cat, five, red' and a message 'If these 5 words were to be sorted alphabetically, the first word to print would be 'cat'.'

Autograded style problems:

For the following problems, write the output of the code. Don't forget `[]` and/or `,` as needed.

- ```
a = 5
b = 'b'
c = True
print("The answer is...", end=' ')
if a != 10:
```

## Exam 1 – MORE Practice Problems

```
 print("A", end=' ')
elif b == 'b':
 print("B", end=' ')
else:
 print("C", end=' ')
z = c and bool(a)
print(z, end=' ')
d = a ** 3 + 25 % 3 - 12 // 5
print(d)
```

2. `mystr = "Howdy! Welcome to Texas A&M Engineering!"`  
`print(mystr[:5] + mystr[6] + mystr[18:] + '\b students!')`

3. `n = 1`  
`p = "A"`  
`while n < 10:`  
    `p += p`  
    `n += 3`  
`print(n, p)`

4. `a = True`  
`b = bool('False')`  
`c = 5 > 8`  
`d = a and b and c`  
`e = not a or not (b and c)`  
`print(d, e)`

5. `mystr = 'The quick brown fox jumped over the lazy dog'`  
`print(mystr[:3], end=' ')`  
`if mystr[4] == 'q':`  
    `if 'fox' in mystr:`  
        `print('fox', end=' ')`  
    `else:`  
        `print('dog', end=' ')`  
    `if mystr[-5] != 'z':`  
        `print('jumped', end=' ')`  
    `else:`  
        `print('hopped', end=' ')`  
`elif 'x' in mystr:`  
    `if 'white' in mystr:`  
        `print('white mouse', end=' ')`  
    `else:`  
        `print('brown cow', end=' ')`  
    `if 'y' not in mystr:`  
        `print('sat', end=' ')`  
    `else:`  
        `print('dropped', end=' ')`  
`else:`  
    `print(mystr[4:26], end=' ')`  
`print('down')`

## Exam 1 – MORE Practice Problems

```
6. mystrs = ['Good Bull', 'Whoop', 'Hullabaloo', 'Howdy', "Gig 'em", 'Aggies']
 mynums = [3, 5, 4, 1, 2]
 for num in mynums:
 print(mystrs[num], end=' ')
```

```
7. mylist = []
 for i in range(5):
 mylist.append(i ** 2)
 print(mylist[-3:])
```

```
8. mydict = {'Ann' : 18, 'Bob' : 20, 'Charlie' : 19}
 if 'Joe' in mydict:
 print("Joe is here")
 elif 'Ann' in mydict:
 print("Hi Ann")
 else:
 print("Anyone?")
```

9. What is the output from the following code:

```
x = 4
y = "Gig'em Aggies!"
while x<100:
 print(x, y)
 x *= (x -2)
 y += y
```

10. What is the output from the following code:

```
x = 5
sum=0
for i in range(4):
 x *= i
 sum += x
print(x, sum)
```

11. What is the output from the following code:

```
AB=0
V=[9, 5, -3, 6, -1, 0]
for i in range(len(V)-2):
 if V[i] < 0:
 AB += 1
print('AB =', AB)
```

12. What is the value of z

```
x = 4
y = 8
t = x
x = y
y = t
z = x / y
print(z)
```

13. Please review all lecture examples and quizzes. Review zyBook: examples, Participation and Challenge Activities.

**Short answer problems:**

1. Briefly explain when it is a good idea to use an `if-elif-else` statement instead of multiple `if` statements.
2. Name 3 reasons for including comments when programming.
3. Briefly explain why it is bad practice to use the “arch” method of program development.
4. Briefly explain when it is best to use a `for` loop vs a `while` loop.
5. List 4 data types we have used in this class so far.
6. What are the differences between lists and dictionaries?
7. What are the similarities between strings and lists?
8. Given the string below, write one line of code to convert it into a list of its words.  
`mystr = 'Aggie Engineers Rock And Are In High Demand By Industry'`
9. Please review all lecture examples and quizzes. Review zyBook: examples, Participation and Challenge Activities.