

ENGR 102 – Fall 2024
Exam 1 Practice Problems

The following problems are for practice when studying for Exam 1. It is recommended that you attempt them using pencil and paper (NOT an IDE) like you will for the exam. After attempting a problem, check your solution by typing it into your favorite IDE and debug.

Partial credit will be available for code writing problems so **please comment your code**. It is recommended that you write out your algorithm in comments first, then go back and fill it in with code (see the “pyramid” method in Lecture 5). Several problems (multiple choice, true/false, fill in the blank, etc) will be autograded and no partial credit will be available.

During the exam calculators are not allowed, you won’t need one anyway. In addition, you may NOT use your phone, the web to search for additional information, your laptop, your book, your notes, lectures on Canvas, or any form of electronic media.

Exam 1 Practice Problems

Autograded style problems:

Go back and review your quizzes for additional autograded style problems including fill in the blank, multiple choice, multiple answer, and true/false type questions.

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

1.

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

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

```
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)
```
4.

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

```
mylist = []
for i in range(5):
    mylist.append(i ** 2)
print(mylist[-3:])
```
6.

```
x = 5 / 5
print(x)
```

Exam 1 Practice Problems

7.

```
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')
```
8.

```
mystr = "Howdy! Welcome to Texas A&M Engineering!"
print(mystr[:5] + mystr[6] + mystr[-22:-1] + ' students!')
```
9.

```
x = 4
y = "Gig'em Aggies!"
while x < 100:
    print(x, y)
    x *= x - 2
    y += y
```
10.

```
x = 5
sum = 0
for i in range(4):
    x *= i
    sum += x
print(x, sum)
```
11.

```
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)
```

Exam 1 Practice Problems

12.

```
x = 4
y = 8
t = x
x = y
y = t
z = x / y
print(z)
```
13.

```
x = 5 % 2 == 1 and 5 < 2 + 4
print(x)
```
14.

```
a = 1
b = 2
c = "a"
d = int(float("3.14"))
if a == 1 and d == 3.14:
    print("Green")
elif c == a or d > 3:
    print("Red")
else:
    print("Yellow")
```
15.

```
x = 10
y = 5
if x % 2 == 0:
    if y > 5:
        print("A")
    else:
        print("B")
        print("C")
else:
    if y < 5:
        print("D")
    else:
        print("E")
        print("F")
print("G")
```
16.

```
x = 2
y = "A"
while x < 100:
    print(x, y)
    x *= x
    y += y
```
17.

```
for i in range(12):
    if i % 2 != 1 and i % 3 == 0:
        print(i)
```

Exam 1 Practice Problems

18. `data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]`
`print(data[1][0][0])`

19. Which code snippet below correctly finds the number of digits of the sum of two integers? Either value may be positive, negative, or zero.

Examples:

$12 + 6 = 18 \rightarrow 2$ digits

$-12 + 2 = -10 \rightarrow 2$ digits (the negative sign does not count)

$0 + 3 = 3 \rightarrow 1$ digit

The following code is executed before each answer below:

```
num1 = int(input("Enter first integer: "))
num2 = int(input("Enter second integer: "))
numSum = num1 + num2
```

- a.

```
myString = str(numSum)
strLength = len(myString) - 1
print(f"number of digits of {numSum} is {strLength}")
```
- b.

```
count = 0
temp = abs(numSum)
while temp > 1:
    temp = temp / 10
    count = count + 1
print(f"number of digits of {numSum} is {count}")
```
- c.

```
if numSum < 0:
    myString = str(-numSum)
else:
    myString = str(numSum)
strLength = len(myString)
print(f"number of digits of {numSum} is {strLength}")
```
- d. None of the above

20. Will the code below run without an error? If not, find and correct the error.

```
side = input("Please enter the side of a square: ")
a = side ** 2
print(f"The area of the square is: {a}")
```

21. Given $x = 3$ and $y = 5$, evaluate the following Boolean expressions:

```
x != y - 2
x >= 0 and not x < 10
x < 0 and x < 10
x >= 0 and x < 2
x < 0 or y < 5
not x > 0 or x < 10
```

Code writing problems:

1. 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 lists of lists.

Example output (input in bold, red text):

```
User 1 please enter a birthday: December 12
User 2 please enter a birthday: January 15
User 3 please enter a birthday: April 12
User 4 please enter a birthday: November 25
User 5 please enter a birthday: April 1
```

```
-----
January 15
April 1
April 12
November 25
December 12
```

2. 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 that is not in the secret word. At the end of the program, print out the number of guesses and the secret word.

Example output (input in bold, red text):

```
Enter the secret word: programming
Guess a letter: n
Guess another letter: a
Guess another letter: e
The secret word is: "programming". You took 3 guesses!
```

3. 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 list of lists and you may assume that it is already available in the code. An example of its data is shown below.

```
roster = [['123004567', ['Joe', 'Aggie', 'ENGE', 3.50]],
          ['123004568', ['Jake', 'Green', 'OCEN', 3.75]],
          ['123004569', ['Jill', 'Apple', 'ENGR', 3.25]]]
```

Example output for input **123004567**:

```
Enter a UIN: 123004567
Joe Aggie: ENGE, 3.50
```

4. Write a Python program that prints out the sum of the even numbers between 2 to 200, inclusive. You must use a loop.

Exam 1 Practice Problems

5. Write a Python program to generate the following patterns exactly as shown using a single loop for each.

a	>	***	oooo	xxxx
bb	>>	**	oooo	xxxo
ccc	>>>	*	ooo	xxoo
dddd	>>	**	oo	xooo
eeee	>	***	o	oooo

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.

Example output (input in bold, red text):

Enter an age: **17**




Enter another age: **24**

...

Enter another age: **-1**

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 convert the number to 800-463-3339. You may assume that the last seven characters are alphabetic characters from A to Z.

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

Example output for input **800-GOFEDEX**:

Enter a phone number in this format XXX-XXXXXXX: **800-GOFEDEX**

800-GOFEDEX is equivalent to 800-463-3339

Exam 1 Practice Problems

8. Write a Python program 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 good coding practices.

Example output using input values **15**, **157**, **3468**, and **12345**:

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.

9. Write a Python program that takes as input a value of n (n is a positive integer) and then calculates and prints 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 $345 + 345345 + 345345345 = 345691035$.

Example output using input values **1**, **12**, and **345**:

Enter an integer: **1**

$1 + 11 + 111 = 123$

Enter an integer: **12**

$12 + 1212 + 121212 = 122436$

Enter an integer: **345**

$345 + 345345 + 345345345 = 345691035$

10. Write a Python program that takes as input a word or sentence and prints the reverse. You **MUST** use a for loop.

Example output for input **howdy all!**:

Enter some text: **howdy all!**

Reversed: !lla ydwoh

Exam 1 Practice Problems

11. Write a Python program to print the table shown below. For each integer n between 2 and 5 (inclusive), print the numbers between n and $n * 10$ that are multiples of n . You MUST use nested loops.

Example output:

```
-----  
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  
5      5, 10, 15, 20, 25, 30, 35, 40, 45, 50  
-----
```

12. Write a Python program that takes as input a positive integer then adds and prints all of the digits in the number.

Example output for input values **12** and **8675309**:

Enter a positive integer: **12**

Sum of the digits: 3

Enter a positive integer: **8675309**

Sum of the digits: 38

13. Write a Python program that takes as input a positive integer that contains at least 10 digits, and a single digit. Have your program remove that digit from the initial number and print the result.

Example output for input values **3479734103487314** and **3**:

Enter an integer with 10+ digits: **3479734103487314**

Enter a digit: **3**

New number is 479741048714

14. Write a Python program that takes as input 5 items that are sold in a school cafeteria (name and cost). Then take as input the amount of money that the user has. Print all of the items that the user can afford to buy.

Example output (input in bold, red text):

Enter 5 items and their cost

Item 1: **apple 0.50**

Item 2: **banana 0.50**

Item 3: **milk 1.00**

Item 4: **pizza 5.00**

Item 5: **hot dog 3.00**

How much money to you have? **2.00**

You can afford apple, banana, or milk

Exam 1 Practice Problems

15. Write a Python program that takes as input a positive integer and then prints all members of the [Fibonacci sequence](#) up to that number (inclusive).

Example output for input **102**:

```
Enter a positive integer: 102
Here is the Fibonacci sequence up to 102:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
```

Now write a Python program that takes as input a positive integer n and then prints the first n members of the [Fibonacci sequence](#).

Example output for input **8**:

```
Enter a positive integer: 8
Here are the first 8 members of the Fibonacci sequence:
0, 1, 1, 2, 3, 5, 8, 13
```

16. Write a Python program that takes as input the number of rows and columns of a 2D $m \times n$ matrix, and prints a list of lists of that matrix. The values of the matrix are the sum of the row and column indices, where indices start at zero. Do NOT use `numpy` or `sympy`. You MUST use a loop.

Example output for input values **3** and **4**:

```
Enter the number of rows: 3
Enter the number of columns: 4
[[ 0, 1, 2, 3], [1, 2, 3, 4], [2, 3, 4, 5]]
```

17. Write a Python program that takes as input a list of numeric values then outputs the second largest value. You may assume that all values are unique. As a challenge, do NOT use the `max()`, `min()`, or `sort()` functions.

Example output (input in bold, red text):

```
Enter some numbers: 1.2 3.4 5.6 7.8 123.456 102 8 6 7 5 3 0 9 -987.6
The second largest value is 102.0
```

18. Write a Python program that will ask the user to input words until the user inputs `stop`, `Stop`, `STOP`, `StOp`, etc. You may assume that all words start with different letters. Have your program print the number of words inputted by the user and the first word if they were arranged in alphabetical order. You may NOT use containers such as lists, dictionaries, sets, or tuples.

Example output (input in bold, red text):

```
Enter a word: dog
Enter another word: howdy
Enter another word: cat
Enter another word: five
Enter another word: red
Enter another word: stoP
If these 5 words were to be sorted alphabetically, the first word
would be "cat"
```

Exam 1 Practice Problems

19. Write a Python program that asks the user to input 5 integers, all on one line, with a single space in between. The program should check to see if any of the 5 numbers are duplicates of another (i.e. check whether any of the integers were entered more than once.) If a duplicate is found, the program should print “Duplicates”, otherwise it should print “All Unique”.

Example output for inputs **1 2 3 3 5**:

```
Enter five integers: 1 2 3 3 5
Duplicates
```

Example output for inputs **1 2 3 4 5**:

```
Enter five integers: 1 2 3 4 5
All Unique
```

20. Write a Python program that will ask the user to input two integers and calculate the sum of the integers between the inputted numbers (inclusive) that are multiples of 4. If the user enters a second integer that is smaller than the first, print a message and do no calculations. Do NOT use containers such as lists, tuples, dictionaries, or sets.

Example output for inputs **2** and **12**:

```
Enter integer 1: 2
Enter integer 2: 12
The sum of multiples of 4 between 2 and 12 is: 24
```

21. Write a Python program that will repeatedly ask a user to enter names and ages of people, stopping when an age of 0 is entered (and not processing that person). The program should collect this information, and then output the average age, the name of the oldest person, and the name of the youngest person. You may assume no two people have the same age.

Example output (input in bold, red text):

```
Enter the name and age of the next person: Ritchey 39
Enter the name and age of the next person: Zoe 6
...
Enter the name and age of the next person: Nobody 0
The average age is 21.3
Frank is the oldest at 85
Ada is the youngest at 3
```

22. Write a Python program that takes as input positive numbers until a negative value is entered. The program should then output the maximum number, the minimum number, and the average value. Do NOT use containers such as lists, tuples, dictionaries, or sets.

Example output (input in bold, red text):

```
Enter a number: 1.1
...
Enter a number: -12
Maximum: 102.0, Minimum: 0.12, Average: 20.25
```

Exam 1 Practice Problems

23. Write a Python program that asks the user for an area, then prints out the radius of a circle with that area, and the length of one side of a square with the same area. Format your output to display one decimal place.

Example output for input **5**:

```
Enter an area: 5
A circle with area 5.0 has radius: 1.3
A square with area 5.0 has side length: 2.2
```

24. Write a Python program that allows the user to enter two (2) integers and then prints all of the values between (and including) the starting and ending integers, that are multiples of both 5 and 7. Format your output nicely with a comma and space between each number.

Example output for inputs **240** and **385**:

```
Enter the first integer: 240
Enter the second integer: 385
Multiples: 245, 280, 315, 350, 385
```

25. Given a list of words stored in the variable `list_words`, write a Python program to print the longest word in the list and its length. You may assume that there is only one word of the longest length in the list.

Example output:

```
The longest word "antidisestablishmentarianism" has 28 characters
```

26. Write a Python program that takes as input a sentence and prints the sentence with every word reversed.

Example output for input **a man a plan a canal panama**:

```
Enter a sentence: a man a plan a canal panama
Words reversed: a nam a nalp a lanac amanap
```

27. The series expansion for $\ln((1+x)/(1-x))$ on the interval $-1 < x < 1$ is as follows:

$$\sum_{n=1}^{\infty} \frac{2}{(2n-1)} x^{2n-1} = 2x + \frac{2}{3}x^3 + \frac{2}{5}x^5 + \frac{2}{7}x^7 + \dots$$

Write a Python program that takes as input a value of x on the interval $-1 < x < 1$. Have your program check that x is within the specified interval, and continue to prompt the user to enter a value until it is. Then compute an approximation for $\ln((1+x)/(1-x))$ using the series expansion summation above. Continue the summation until the absolute value of the term to be added is less than 10^{-6} . For example, if $x = 0.5$ the first term for $n = 1$ is $\frac{2}{(2*1-1)} 0.5^{2*1-1}$ or 1. Since this term is greater than 10^{-6} , add the term to the summation and continue. Eventually one of the terms will be less than 10^{-6} , and the summation stops and prints the result.

Example output for input **0.5**:

```
Enter a value for x: 0.5
ln((1+x)/(1-x)) is approximately 1.098611131435838
```

Exam 1 Practice Problems

28. The Maclaurin series expansion for $1/(1 - x)$ on the interval $-1 < x < 1$ is as follows:

$$\sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + x^4 + \cdots + x^n$$

Write a Python program that takes as input a value of x on the interval $-1 < x < 1$ then computes an approximation for $1/(1 - x)$ using the series expansion summation above. The summation should continue until the term to be added is less than 10^{-6} in absolute value. Hint: Note that each term in the series is x raised to a power, including the first two terms: $x^0 = 1$ and $x^1 = x$.

Example output for input **0.5**:

Enter a value for x: **0.5**

`1/(1-0.5) is approximately 1.9999980926513672`

29. Given a list `xdata` of arbitrary length that contains values of x , write a Python program to calculate a y value for each x value using the equation below. Store the calculated y values in a list named `ydata`. Do NOT use `numpy` or `sympy`. Use list functions, methods, and operators.

$$y = 4.12x^2 + 1.52x - 7.1$$

30. Sheldon Cooper's (of Big Bang Theory) favorite number is 73. One of the reasons is that 73 is a prime number and there are 21 prime numbers between 1 and 73. A prime number is an integer greater than 1 that is not divisible by another integer other than 1 (the only even number that is a prime number is 2; all other prime numbers are odd). Write a Python program to calculate and print the prime numbers between 1 and 73 (but not including 73). Your program will also need to count the prime numbers to see if this is really the 21st prime number.
31. Write a Python program that takes as input an arbitrary number of masses and corresponding volumes then calculates and prints the density for each pair.

Example output (input in bold, red text):

Enter the masses: **10 22.5 30**

Enter the volumes: **5 2 1.5**

The densities are: 2.0, 11.25, 20.0

32. Write a Python program that takes as input a combination lock (a 4-digit integer) then "solve" the combination. Practice using lists of lists to store the lock.
33. Write a Python program that takes as input the radius of a circle then calculates and prints the items below. Print the values using two decimal places.
- The area and perimeter (circumference) of the circle
 - The side length of a square with the same area as the circle
 - The side length of a square with the same perimeter as the circle

Example output for input **1.0**:

Enter the radius: **1.0**

The circle has area 3.14 and perimeter 6.28

A square with equal area has side length 1.77

A square with equal perimeter has side length 1.57

Short answer problems for studying:

1. What are the differences in the following mathematical operators? `/`, `%`, `//`
2. How do you format your output to display a number with exactly 3 decimal places?
3. What are the different assignment operators? Provide an example for each.
4. List all of the data types we have used in this class so far and provide an example for each. How do you convert between data types?
5. Briefly explain when it is a good idea to use an `if-elif-else` statement instead of multiple `if` statements. When is it a good idea to nest `if` statements?
6. Name 3 good reasons for including comments when programming.
7. Briefly explain why it is bad practice to use the “arch” method of program development. Briefly explain the “pyramid” approach to program development.
8. What is “debugging”?
9. Briefly explain when it is best to use a `for` loop vs a `while` loop.
10. What are the similarities between strings and lists?
11. 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'`
12. Given a list `L` of length greater than 10, write the code to create a new list `L_new` containing the 4th through 7th elements of `L`. Next write the code to remove the 2nd, 3rd, and 4th elements of `L`. Next write code to insert a new value as the 2nd element of `L`.
13. Please review all lecture examples and quizzes. Also review zyBook examples as well as Participation (orange) and Challenge (blue) activities.