

Double-click (or enter) to edit

Program 81 Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. Suppose the following input is supplied to the program

```
input_sentence = input("Enter a sentence: ")
# Split the sentence into words
words = input_sentence.split()
# Create a dictionary to store word frequencies
word_freq = {}
# Count word frequencies
for word in words:
    # Remove punctuation (., ?) from the word
    word = word.strip('.,?')
    # Convert the word to lowercase to ensure case-insensitive counting
    word = word.lower()
    # Update the word frequency in the dictionary
    if word in word_freq:
        word_freq[word] += 1
    else:
        word_freq[word] = 1
# Sort the words alphanumerically
sorted_words = sorted(word_freq.items())
# Print the word frequencies
for word, frequency in sorted_words:
    print(f"{word}:{frequency}")
```

```
Enter a sentence: New to python or choosing between python 2 and python 3 ? Read python 2 or python 3
:1
2:2
3:2
and:1
between:1
choosing:1
new:1
or:2
python:5
read:1
to:1
```

Program 82 Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

```
class Person:
    def getGender(self):
        return "Unknown"
class Male(Person):
    def getGender(self):
        return "Male"
class Female(Person):
    def getGender(self):
        return "Female"
person = Person()
male = Male()
female = Female()
print(person.getGender())
print(male.getGender())
print(female.getGender())
```

```
➞ Unknown
   Male
   Female
```

Program 83 Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey","Football"].

```
subjects = ["I", "You"]
verbs = ["Play", "Love"]
objects = ["Hockey", "Football"]
sentences = []
for sub in subjects:
    for vrb in verbs:
        for obj in objects:
            sentence = f"{sub} {vrb} {obj}."
            sentences.append(sentence)
for sentence in sentences:
    print(sentence)
```

```
➞ I Play Hockey.
   I Play Football.
```

```

I Love Hockey.
I Love Football.
You Play Hockey.
You Play Football.
You Love Hockey.
You Love Football.

```

Program 84 Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

```

import zlib
string = "hello world!hello world!hello world!hello world!"
# Compress the string
compressed_string = zlib.compress(string.encode())
# Decompress the string
decompressed_string = zlib.decompress(compressed_string).decode()
print("Original String:", string)
print("Compressed String:", compressed_string)
print("Decompressed String:", decompressed_string)

```

```

➦ Original String: hello world!hello world!hello world!hello world!
Compressed String: b'x\x9c\xcbH\xcd\x9W(\xcf/\xcaIQ\xcc \x82\r\x00\xbd[\x11\xf5'
Decompressed String: hello world!hello world!hello world!hello world!

```

Program 85 Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list

```

def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid # Element found, return its index
        elif arr[mid] < target:
            left = mid + 1 # Target is in the right half
        else:
            right = mid - 1 # Target is in the left half
    return -1 # Element not found in the list

# Example usage:
sorted_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
target_element = 4
result = binary_search(sorted_list, target_element)

```

```

if result != -1:
    print(f"Element {target_element} found at index {result}")
else:
    print(f"Element {target_element} not found in the list")

```

➞ Element 4 not found in the list

Program 86 Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console. Example: If the following n is given as input to the program: 100 Then, the output of the program should be: 0,35,70

```

def divisible_by_5_and_7(n):
    for num in range(n + 1):
        if num % 5 == 0 and num % 7 == 0:
            yield num
try:
    n = int(input("Enter a value for n: "))
    result = divisible_by_5_and_7(n)
    print(','.join(map(str, result)))
except ValueError:
    print("Invalid input. Please enter a valid integer")

```

➞ Enter a value for n: 264  
0,35,70,105,140,175,210,245

Program 87 Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console. Example: If the following n is given as input to the program: 10 Then, the output of the program should be: 0,2,4,6,8,10

```

def even_numbers(n):
    for num in range(n + 1):
        if num % 2 == 0:
            yield num
try:
    n = int(input("Enter a value for n: "))
    result = even_numbers(n)
    print(','.join(map(str, result)))
except ValueError:
    print("Invalid input. Please enter a valid integer for n.")

```

Enter a value for n: 50  
 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50

### Program 88 Program for the fibonacci sequence using list comprehension

```
def fibonacci(n):
    sequence = [0, 1] # Initializing the sequence with the first two F
    [sequence.append(sequence[-1] + sequence[-2]) for _ in range(2, n)]
    return sequence
try:
    n = int(input("Enter a value for n: "))
    result = fibonacci(n)
    print(','.join(map(str, result)))
except ValueError:
    print("Invalid input. Please enter a valid integer for n.")
```

Enter a value for n: 10  
 0,1,1,2,3,5,8,13,21,34

Program 89 Assuming that we have some email addresses in the "[username@companyname.com](mailto:username@companyname.com)" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

```
def extract_username(email):
    # Split the email address at '@' to separate the username and domai
    parts = email.split('@')
    # Check if the email address has the expected format
    if len(parts) == 2:
        return parts[0] # The username is the first part
    else:
        return "Invalid email format"
try:
    email = input("Enter an email address: ")
    username = extract_username(email)
    print(username)
except ValueError:
    print("Invalid input. Please enter a valid email address.")
```

Enter an email address: [amitvsuryavanshi04@gmail.com](mailto:amitvsuryavanshi04@gmail.com)  
 amitvsuryavanshi04



Program 90 Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape's area is 0 by default.

Program 90 Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape's area is 0 by default.

```
class Shape:
    def __init__(self):
        pass # Default constructor, no need to initialize anything

    def area(self):
        return 0 # Shape's area is 0 by default

class Square(Shape):
    def __init__(self, length):
        super().__init__() # Call the constructor of the parent class
        self.length = length

    def area(self):
        return self.length ** 2 # Calculate the area of the square

# Create instances of the classes
shape = Shape()
square = Square(float(input("Enter the side length of the square: ")))

# Calculate and print the areas
print("Shape's area by default:", shape.area())
print("Area of the square:", square.area())
```

```
Enter the side length of the square: 10
Shape's area by default: 0
Area of the square: 100.0
```

