python-asin-modern

October 16, 2023

1 1. Create a program that asks the user to enter their name and their age. Print out a

message addressed to them that tells them the year that they will turn 100 years old

[1]:
```
name = input("Enter your name: ")
age = int(input("Enter your age: "))
current_year = 2023  # You can update this with the current year
year_turn_100 = current_year + (100 - age)
message = f"Hello, {name}! You will turn 100 years old in the year␣
  ↪{year_turn_100}."
print(message)
```

```
Enter your name: subhasish
Enter your age: 22
Hello, subhasish! You will turn 100 years old in the year 2101.
```

2 2. Write a program to check whether the number is even or odd, print out an

appropriate message to the user.

[2]:
```
num = int(input("Enter the number :"))

if num%2==0 :
    print(f"The {num} is even")
else :
    print(f"The {num} is odd")
```

```
Enter the number :5
The 5 is odd
```

3 3. Write a program which will find all such numbers which are divisible by 7

[3]:

```
start = int(input("Enter the start of the range: "))
end = int(input("Enter the end of the range: "))

divisible_by_7 = []

for number in range(start, end + 1):
    if number % 7 == 0:
        divisible_by_7.append(number)

if divisible_by_7:
    print("Numbers divisible by 7 in the range from" ,start, "to" ,end,":")
    for num in divisible_by_7:
        print(num)
else:
    print(f"There are no numbers divisible by 7 in the range from",start, "to"␣
 ↪,end,".")
```

```
Enter the start of the range: 1
Enter the end of the range: 10
Numbers divisible by 7 in the range from 1 to 10 :
7
```

4 4. Write a program which can compute the factorial of a given numbers.

[4]:

```
num = int(input("Enter a number: "))

def facto(n):
    if n == 0:
        return 1
    else:
        return n * facto(n - 1)

if num < 0:
    print("Factorial is not defined for negative numbers.")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    result = facto(num)
    print(f"The factorial of {num} is {result}")
```

```
Enter a number: 5
The factorial of 5 is 120
```

5 5. Write a program that prints out all the elements of the list that are less than 10.

[6]:

```python
number =[2,33,5,6,44,7,34,8,34,9,12,10,11]
print("Element less than 10 are :")
for num in number:
    if num<10:
        print(num)
```

```
Element less than 10 are :
2
5
6
7
8
9
```

6 6. Write a program that returns a list that contains only the elements that are

common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

[8]:

```python
l1=[333,444,55,76,98,33,7876,44,55]
l2=[22,44,5,367,77,556,78,54,77,44]
if len(l1)==len(l2):
    print("list have samelength")
else:
    print("List do not have same length ")
    print(set(l1).intersection(l2))
```

```
List do not have same length
{44}
```

7 7. To determine whether the number is prime or not.

[13]:

```python
num = 29
flag = False
if num == 1:
    print(num, "is not a prime number")
elif num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            flag = True
            break
    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

```
29 is a prime number
```

8 8. To check whether a number is palindrome or not.

[22]:

```
num=696
snum=str(num)
rev=snum[::-1]
if snum==rev:
    print("Its a palindrom")
else:
    print("Its not palindrom")
```

```
its a palindrom
```

9 9. Write a program that asks the user how many Fibonnaci numbers to generate and

then generates them.

[23]:

```
def generate_fibonacci(n):
    fibonacci_sequence = []

    if n <= 0:
        return fibonacci_sequence

    a, b = 0, 1
    fibonacci_sequence.append(a)

    for _ in range(1, n):
        a, b = b, a + b
        fibonacci_sequence.append(a)

    return fibonacci_sequence

num_terms = int(input("Enter the number of Fibonacci numbers to generate:"))

fibonacci_numbers = generate_fibonacci(num_terms)

if len(fibonacci_numbers) == 0:
    print("No Fibonacci numbers to generate.")
else:
    print("Generated Fibonacci numbers:")
    for num in fibonacci_numbers:
        print(num)
```

```
Enter the number of Fibonacci numbers to generate:9
Generated Fibonacci numbers:
0
1
1
2
3
5
8
13
21
```

10 10. Write a program (using functions!) that asks the user for a long string containing

multiple words. Print back to the user the same string, except with the words in backwards order. E.g “ I am Msc student” is :”student Msc am I”

[24]:
```
inputString = input("Enter you string : ")
print(inputString)
inputString = inputString.split(" ")
print(inputString)
inputString.reverse()
inputString = " ".join(inputString)
print(inputString)
```

```
Enter you string : hey am subhasish karmkar
hey am subhasish karmkar
['hey', 'am', 'subhasish', 'karmkar', '']
 karmkar subhasish am hey
```

11 11. Write a program to implement binary search to search the given element using

function.

[59]:
```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1

    while left <= right:
        mid = (left + right) // 2

        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1

sorted_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
search_element = int(input("Enter the element to search:"))

result = binary_search(sorted_list, search_element)

if result != -1:
    print(f"The element {search_element} is found at index {result}.")
else:
    print(f"The element {search_element} is not found in the list.")
```

```
Enter the element to search:4555
The element 4555 is not found in the list.
```

12 12. Given a .txt file that has a list of a bunch of names, count how many of each

name there are in the file, and print out the results to the screen.

[26]:

```
# Open the file in read mode
with open("names.txt", "r") as file:
    # Read names from the file and split them into a list
    names_list = file.read().splitlines()

# Create a dictionary to store name counts
name_counts = {}

# Count occurrences of each name
for name in names_list:
    name_counts[name] = name_counts.get(name, 0) + 1

# Print the results
print("Name Counts:")
for name, count in name_counts.items():
    print(f"{name}: {count}")
```

```
Name Counts:
subhasish: 1
sohel : 1
shnatanu: 1
nayan : 1
```

13 13. Write a program that takes a list of numbers (for example, $a = [5, 10, 15, 20, 25]$)

and makes a new list of only the first and last elements of the given list and makes a new list of only the first.

[33]:

```
c = [5, 10, 15, 20, 25]
print (c)
def firstLast (a):
    new_list = [(a[0],a[-1])]
    print (new_list)
firstLast(c)
```

```
[5, 10, 15, 20, 25]
[(5, 25)]
```

14 14. Write a program that accepts sequence of lines as input and prints the lines after

making all characters in the sentence capitalized.

[41]:

```
l = input("Enter a line (or press Enter to finish): ")
c_line=l.upper()
print(l)
print("Capitalized :",c_line)
```

```
Enter a line (or press Enter to finish): subhaish
subhaish
Capitalized : SUBHAISH
```

15 15. Write a program that accepts a sentence and calculate the number of letters and digits.

[43]:

```
s = input("Input a string: ")
d=l=0
for c in s:
    if c.isdigit():
        d=d+1
    elif c.isalpha():
        l=l+1
    else:
            pass
print("Letters", l)
print("Digits", d)
```

```
Input a string: shantanu pandit 9422
```

```
Letters 14
Digits 4
```

16 16. Write a program that accepts a sentence and calculate the number of upper case

letters and lower case letters

[46]:

```python
sentence = input("Enter a sentence: ")
uppercase_count = 0
lowercase_count = 0
for char in sentence:
    if char.isupper():
        uppercase_count += 1
    elif char.islower():
        lowercase_count += 1
print("Uppercase letters:",uppercase_count)
print("Lowercase letters:",lowercase_count)
```

```
Enter a sentence: SUBHASISH karmakar
Uppercase letters: 9
Lowercase letters: 8
```

17 17. Write a Python function to calculate the factorial of a number (a non-negative

integer). The function accepts the number as an argument.

[50]:

```python
def factorial(n):
    if n < 0:
        return "Factorial is undefined for negative numbers"
    elif n == 0:
        return 1
    else:
        return n * factorial(n - 1)

num = int(input("Enter a non-negative integer: "))
result = factorial(num)

if type(result) == int:
    print(f"The factorial of {num} is {result}.")
else:
    print(result)
```

```
Enter a non-negative integer: 4
The factorial of 4 is 24.
```

18 18:Write a Python program to count the number of lines in a text file.

[51]:

```
file_path = 'names.txt'
    with open(file_path, 'r') as file:
        line_count = sum(1 for line in file)
    print(f"Number of lines in the file: {line_count}")
except FileNotFoundError:
    print(f"File '{file_path}' not found.")
except Exception as e:
    print(f"An error occurred: {str(e)}")
```

```
Number of lines in the file: 4
```

19 19. Write a Python program to copy the contents of a file to another file

[52]:

```
source_file_path = 'names.txt'
destination_file_path = 'dup.txt'
try:
    with open(source_file_path, 'r') as source_file:
        file_contents = source_file.read()

    with open(destination_file_path, 'w') as destination_file:
        destination_file.write(file_contents)

    print(f"Contents copied from '{source_file_path}' to␣
 ↪'{destination_file_path}'.")

except FileNotFoundError:
    print("File not found.")
except Exception as e:
    print(f"An error occurred: {str(e)}")
```

```
Contents copied from 'names.txt' to 'dup.txt'.
```

20 20. Write a Python program to solve the Fibonacci sequence using recursion.

[54]:

```
def fibbonci(n):
    if n in {0,1}:
        return n
    else:
        return (fibbonci(n-1)+fibbonci(n-2))
no=int(input("Enter end number upto which fibbonci series to print: "))
```

```
for i in range(0,no,1):
    print(fibbonci(i))
```

```
Enter end number upto which fibbonci series to print: 5
0
1
1
2
3
```

21 21. Write a Python class named Circle constructed by a radius and two methods

which will compute the area and the perimeter of a circle.

[58]:

```
import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return math.pi * self.radius**2

    def perimeter(self):
        return 2 * math.pi * self.radius

# Example usage:
radius = float(input("Enter the radius of the circle: "))
my_circle = Circle(radius)

circle_area = my_circle.area()
circle_perimeter = my_circle.perimeter()

print(f"Radius: {my_circle.radius}")
print(f"Area: {circle_area:.2f}")
print(f"Perimeter: {circle_perimeter:.2f}")
```

```
Enter the radius of the circle: 5
Radius: 5.0
Area: 78.54
Perimeter: 31.42
```

[ ]: