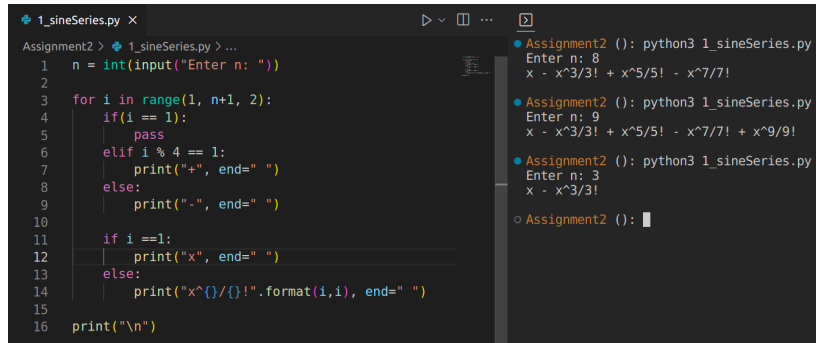


Programming in Python CST 362

Assignment 2

Mayon Francis
CS6A
44

1. Print the sin series $x - x^3/3! + x^5/5! - \dots x^n/n!$ (read n)



```
1_sineSeries.py
Assignment2 > 1_sineSeries.py > ...
1 n = int(input("Enter n: "))
2
3 for i in range(1, n+1, 2):
4     if(i == 1):
5         pass
6     elif i % 4 == 1:
7         print("+", end=" ")
8     else:
9         print("-", end=" ")
10
11     if i == 1:
12         print("x", end=" ")
13     else:
14         print("x^{}/{}!".format(i,i), end=" ")
15
16 print("\n")
```

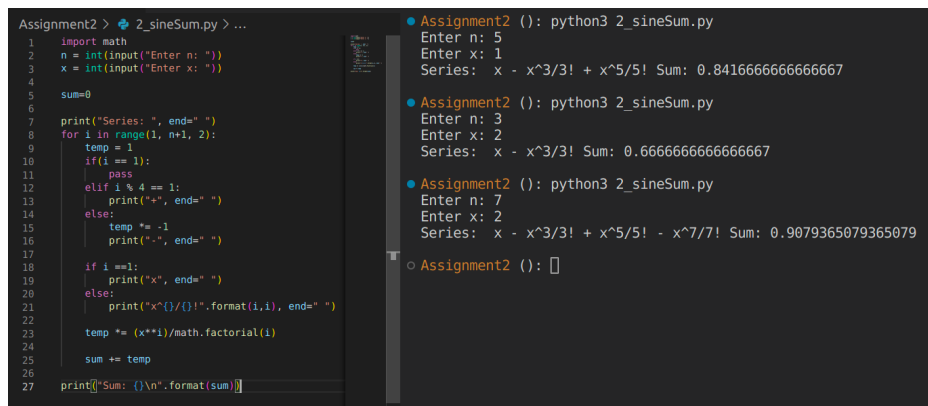
Assignment2 (): python3 1_sineSeries.py
Enter n: 8
x - x^3/3! + x^5/5! - x^7/7!

Assignment2 (): python3 1_sineSeries.py
Enter n: 9
x - x^3/3! + x^5/5! - x^7/7! + x^9/9!

Assignment2 (): python3 1_sineSeries.py
Enter n: 3
x - x^3/3!

Assignment2 ():

2. In the above program read the value x and find the sum of the series



```
Assignment2 > 2_sineSum.py > ...
1 import math
2 n = int(input("Enter n: "))
3 x = int(input("Enter x: "))
4
5 sum=0
6
7 print("Series: ", end=" ")
8 for i in range(1, n+1, 2):
9     temp = 1
10    if(i == 1):
11        pass
12    elif i % 4 == 1:
13        print("+", end=" ")
14    else:
15        temp *= -1
16        print("-", end=" ")
17
18    if i == 1:
19        print("x", end=" ")
20    else:
21        print("x^{}/{}!".format(i,i), end=" ")
22
23    temp *= (x**i)/math.factorial(i)
24    sum += temp
25
26 print("\nSum: {}\\n".format(sum))
```

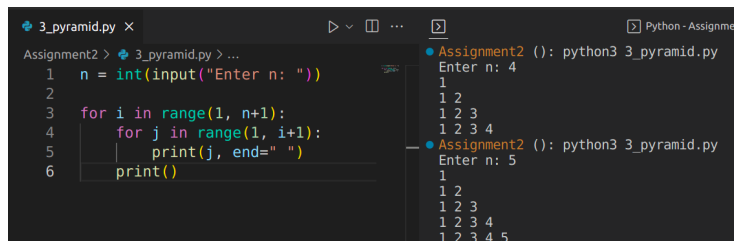
Assignment2 (): python3 2_sineSum.py
Enter n: 5
Enter x: 1
Series: x - x^3/3! + x^5/5! Sum: 0.8416666666666667

Assignment2 (): python3 2_sineSum.py
Enter n: 3
Enter x: 2
Series: x - x^3/3! Sum: 0.6666666666666667

Assignment2 (): python3 2_sineSum.py
Enter n: 7
Enter x: 2
Series: x - x^3/3! + x^5/5! - x^7/7! Sum: 0.9079365079365079

Assignment2 ():

3. Print pyramid for n

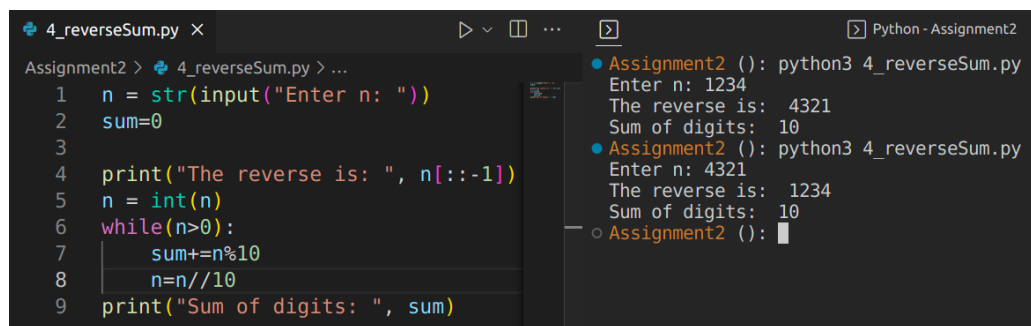


```
3_pyramid.py
Assignment2 > 3_pyramid.py > ...
1 n = int(input("Enter n: "))
2
3 for i in range(1, n+1):
4     for j in range(1, i+1):
5         print(j, end=" ")
6     print()
```

Assignment2 (): python3 3_pyramid.py
Enter n: 4
1
1 2
1 2 3
1 2 3 4

Assignment2 (): python3 3_pyramid.py
Enter n: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

4. Reverse a number and also find the sum of the digits



```
4_reverseSum.py
Assignment2 > 4_reverseSum.py > ...
1 n = str(input("Enter n: "))
2 sum=0
3
4 print("The reverse is: ", n[::-1])
5 n = int(n)
6 while(n>0):
7     sum+=n%10
8     n=n//10
9 print("Sum of digits: ", sum)
```

Assignment2 (): python3 4_reverseSum.py
Enter n: 1234
The reverse is: 4321
Sum of digits: 10

Assignment2 (): python3 4_reverseSum.py
Enter n: 4321
The reverse is: 1234
Sum of digits: 10

Assignment2 ():

5. Check if armstrong

```
5_armstrong.py x
Assignment2 > 5_armstrong.py > ...
1 num = int(input("Enter a number: "))
2 order = len(str(num))
3 sum = 0
4 for digit in str(num):
5     sum += int(digit) ** order
6
7 if sum == num:
8     print("{} is armstrong".format(num))
9 else:
10    print("{} is not armstrong".format(num))
```

```
• Assignment2 (): python3 5_armstrong.py
Enter a number: 153
153 is armstrong
• Assignment2 (): python3 5_armstrong.py
Enter a number: 152
152 is not armstrong
○ Assignment2 ():
```

6. Calculate Root using newton's method

```
6_sqRootNewton.py x
Assignment2 > 6_sqRootNewton.py > ...
1 n = int(input("Enter a number: "))
2 x = n
3 # To count the number of iterations
4 count = 0
5
6 while (1):
7     count += 1
8     # Calculate more closed x
9     root = 0.5 * (x + (n / x))
10    # Check for closeness
11    if (abs(root - x) < 0.00001):
12        break
13    # Update root
14    x = root
15
16 print("The square root of", n, "is", root)
```

```
• Assignment2 (): python3 6_sqRootNewton.py
Enter a number: 4
The square root of 4 is 2.0000000000000002
• Assignment2 (): python3 6_sqRootNewton.py
Enter a number: 10000
The square root of 10000 is 100.0
• Assignment2 (): python3 6_sqRootNewton.py
Enter a number: 16
The square root of 16 is 4.00000000000000051
• Assignment2 (): python3 6_sqRootNewton.py
Enter a number: 10000000000
The square root of 10000000000 is 100000.0
○ Assignment2 ():
```

7. Investment Report

```
Assignment2 > 7_investReport.py > ...
1 startAmount = int(input("Enter the starting amount: "))
2 interestRate = float(input("Enter the interest rate: "))
3 years = int(input("Enter the number of years: "))
4
5
6 print(f"Year    Starting Balance    Interest    Ending Balance")
7 yearStartAmount=startAmount
8 for i in range(1, years+1):
9     interest = yearStartAmount * interestRate / 100
10    print(f"{i:<8}{yearStartAmount:<20}", end="")
11    print(f"interest:<11.2f}{yearStartAmount+interest:<15.2f}")
12    yearStartAmount += interest
13
14 endAmount = yearStartAmount
15 print(f"Ending Balance: {endAmount:.2f}")
16 print(f"Total interest earned: {endAmount-startAmount:0.2f}")
```

```
• Assignment2 (): python3 7_investReport.py
Enter the starting amount: 10000
Enter the interest rate: 5
Enter the number of years: 5
Year    Starting Balance    Interest    Ending Balance
1       10000              500.00     10500.00
2      10500.0          525.00     11025.00
3      11025.0          551.25     11576.25
4      11576.25        578.81     12155.06
5      12155.0625       607.75     12762.82
Ending Balance: 12762.82
Total interest earned: 2762.82
○ Assignment2 ():
```

8. Check if krishnamurthy number

```
8_krishnamurthy.py x
Assignment2 > 8_krishnamurthy.py > ...
1 import math
2
3 n = str(input("Enter a number: "))
4 sum = 0
5 for d in n:
6     sum += math.factorial(int(d))
7
8 if sum == int(n):
9     print(f"{n} is a Krishnamurthy number.")
10 else:
11    print(f"{n} is not a Krishnamurthy number.")
```

```
• Assignment2 (): python3 8_krishnamurthy.py
Enter a number: 143
143 is not a Krishnamurthy number.
• Assignment2 (): python3 8_krishnamurthy.py
Enter a number: 145
145 is a Krishnamurthy number.
○ Assignment2 ():
```

9. Sum of First and Last digit

```
9_firstLast.py X
Assignment2 > 9_firstLast.py > ...
1 n = input("Enter a number: ")
2
3 sum = int(n[0]) + int(n[-1])
4
5 print("Sum of first and last digit is: ", sum)
```

• Assignment2 (): python3 9_firstLast.py
Enter a number: 123
Sum of first and last digit is: 4
• Assignment2 (): python3 9_firstLast.py
Enter a number: 765
Sum of first and last digit is: 12
○ Assignment2 ():

10. Print number in words (Digit wise)

```
10_numToDigitWord.py X
Assignment2 > 10_numToDigitWord.py > numToDigitWord
1 def numToDigitWord(num):
2     if num == 1:
3         return "One"
4     elif num == 2:
5         return "Two"
6     elif num == 3:
7         return "Three"
8     elif num == 4:
9         return "Four"
10    elif num == 5:
11        return "Five"
12    elif num == 6:
13        return "Six"
14    elif num == 7:
15        return "Seven"
16    elif num == 8:
17        return "Eight"
18    elif num == 9:
19        return "Nine"
20    elif num == 0:
21        return "Zero"
22    else:
23        return "Invalid Number"
24
25 n = input("Enter a number: ")
26
27 for d in n:
28     print(numToDigitWord(int(d)), end=" ")
29
30 print()
```

• Assignment2 (): python3 10_numToDigitWord.py
Enter a number: 345
Three Four Five
• Assignment2 (): python3 10_numToDigitWord.py
Enter a number: 987
Nine Eight Seven
• Assignment2 (): python3 10_numToDigitWord.py
Enter a number: 1234567890
One Two Three Four Five Six Seven Eight Nine Zero
○ Assignment2 ():

Strings

1. String Manipulation

```
11_stringManipulate.py X
Assignment2 > 11_stringManipulate.py > ...
1 # Initializing the string
2 str = "Python Programming by Mayon"
3
4 # a) To display the last four characters.
5 print("Last four characters:", str[-4:])
6 # b) To display the substring starting from index 4 and ending at index 8
7 print("Substring from index 4 to 8:", str[4:9])
8 # c) Find the length of the string, min and max(characters)
9 print("Length of the string:", len(str))
10 print("Minimum character:", min(str))
11 print("Maximum character:", max(str))
12 # d) To trim the last four characters from the string.
13 trimmed_str = str[:-4]
14 print("String after trimming last four characters:", trimmed_str)
15 # e) To trim the first four characters from the string.
16 trimmed_str = str[4:]
17 print("String after trimming first four characters:", trimmed_str)
18 # f) To display the starting index of the substring 'gr'.
19 print("Starting index of the substring 'gr':", str.index("gr"))
20 # g) To change the case of the given string: (small letter to capital and
21 print("String in swapped case:", str.swapcase())
22 # h) To check if the string is in title case.
23 print("Is the string in title case?", str.istitle())
24 # i) To replace all the occurrences of letter 'm' in the string with '*'
25 new_str = str.replace("m", "*")
26 print("String after replacing 'm' with '*':", new_str)
27 # j) Reverse the string
28 reversed_str = str[::-1]
29 print("Reversed string:", reversed_str)
30 # k) Count the occurrence of the character 'm'
31 m_count = str.count('m')
32 print("Number of occurrences of 'm':", m_count)
33 # l) Characters in even positions 0,2,4,...
34 print("Characters in even positions:", str[::2])
35 # m) Characters in even positions 0,2,4,... in reverse order
36 print("Characters in even positions in reverse order:", str[::-2])
37 # n) Check whether the substring 'on' is present in the string or not
38 print("Is 'on' present in the string?", 'on' in str)
39 # o) Find the first occurrence of character 't'
40 print("Index of first occurrence of 't':", str.index('t'))
41 # p) Convert the string into upper case
42 print("Uppercase string:", str.upper())
```

• Assignment2 (): python3 11_stringManipulate.py
Last four characters: ayon
Substring from index 4 to 8: on Pr
Length of the string: 27
Minimum character: y
Maximum character: y
String after trimming last four characters: Python Programming by M
String after trimming first four characters: on Programming by Mayon
Starting index of the substring 'gr': 10
String in swapped case: pYTHON pROGRAMMING BY mAYON
Is the string in title case? False
String after replacing 'm' with '*': Python Progra**ing by Mayon
Reversed string: noyM yb gnimmargorP nohtyP
Number of occurrences of 'm': 2
Characters in even positions: Pto rgamn yMyn
Characters in even positions in reverse order: nyMy nmagr otP
Is 'on' present in the string? True
Index of first occurrence of 't': 2
Uppercase string: PYTHON PROGRAMMING BY MAYON
○ Assignment2 ():

2. Check if palindrome

```
12_palindrome.py x
Assignment2 > 12_palindrome.py > ...
1 n = input("Enter a string: ")
2
3 if n == n[::-1]:
4     print(f"{n} is a palindrome.")
5 else:
6     print(f"{n} is not a palindrome.")
```

- Assignment2 (): python3 12_palindrome.py
Enter a string: hello
hello is not a palindrome.
- Assignment2 (): python3 12_palindrome.py
Enter a string: malayalam
malayalam is a palindrome.
- Assignment2 ():

3. Count

```
13_strCount.py x
Assignment2 > 13_strCount.py > n
1 n = input("Enter a string: ")
2 vowels = 0
3 digits = 0
4 consonents = 0
5 spaces = 0
6 for c in n:
7     if c in "aeiouAEIOU":
8         vowels += 1
9     elif c.isdigit():
10        digits += 1
11    elif c == " ":
12        spaces += 1
13    else:
14        consonents += 1
15
16 print(f"Vowels: {vowels}")
17 print(f"Digits: {digits}")
18 print(f"Consonents: {consonents}")
19 print(f"Spaces: {spaces}")
```

- Assignment2 (): python3 13_strCount.py
Enter a string: Mayon CS6A
Vowels: 3
Digits: 1
Consonents: 5
Spaces: 1
- Assignment2 ():

4. Binary To Decimal

```
13_strCount.py  ~lock,jaison.odt#  14_binToDec.py x
Elective > Assignment2 > 14_binToDec.py > ...
1 binInput = input("Enter a binary number: ")
2 decOutput = 0
3 print("The decimal value of the binary number is: ", int(binInput, 2))
4
5 for b in binInput:
6     if b not in ['0', '1']:
7         print("Invalid binary number")
8         break
9     decOutput = decOutput * 2 + int(b)
10
11 print(f"The decimal value of the binary number is: ", decOutput)
```

- Assignment2 (): python3 14_binToDec.py
Enter a binary number: 1010
The decimal value of the binary number is: 10
The decimal value of the binary number is: 10
- Assignment2 ():

5. Read a decimal number and find its binary.(Hint: divide by 2 and append the remainder to a string)

```
15_decToBinary.py x
Elective > Assignment2 > 15_decToBinary.py > ...
1 # 5. Read a decimal number and find its binary.( Hint: divide by 2 and ar
2
3 n = int(input("Enter a number: "))
4 nCopy = n
5 binary = ""
6 while n > 0:
7     binary = str(n % 2) + binary
8     n = n // 2
9
10 print(f"Binary of {nCopy} is {binary}")
```

- Assignment2 (): python3 15_decToBinary.py
Enter a number: 15
Binary of 15 is 1111
- Assignment2 (): python3 15_decToBinary.py
Enter a number: 2
Binary of 2 is 10
- Assignment2 ():

6. Binary to Hex

```
16_binToHex.py x
```

```
Elective > Assignment2 > 16_binToHex.py > hexConv
```

```
1 def binToHex(bin):
2     hex = ""
3     for i in range(0, len(bin), 4):
4         hex += str(hexConv(bin[i:i+4]))
5     return hex
6
7 def hexConv(bin):
8     if bin == "0000":
9         return 0
10    elif bin == "0001":
11        return 1
12    elif bin == "0010":
13        return 2
14    elif bin == "0011":
15        return 3
16    elif bin == "0100":
17        return 4
18    elif bin == "0101":
19        return 5
20    elif bin == "0110":
21        return 6
22    elif bin == "0111":
23        return 7
24    elif bin == "1000":
25        return 8
26    elif bin == "1001":
27        return 9
28    elif bin == "1010":
29        return "A"
30    elif bin == "1011":
31        return "B"
32    elif bin == "1100":
33        return "C"
34    elif bin == "1101":
35        return "D"
36    elif bin == "1110":
37        return "E"
38    elif bin == "1111":
39        return "F"
40
41 n = input("Enter an 8 bit binary number: ")
42 if len(n) != 8:
43     print("Invalid input")
44 else:
45     print(f"Hexadecimal of {n} is {binToHex(n)}")
46
```

- Assignment2 (): python3 16_binToHex.py
Enter an 8 bit binary number: 10101010
Hexadecimal of 10101010 is AA
- Assignment2 (): python3 16_binToHex.py
Enter an 8 bit binary number: 11111010
Hexadecimal of 11111010 is FA
- Assignment2 (): python3 16_binToHex.py
Enter an 8 bit binary number: 11
Invalid input
- Assignment2 ():

7. Hex to Decimal and Binary

```
17_hexToBinDec.py x
```

```
Elective > Assignment2 > 17_hexToBinDec.py > ...
```

```
1 n = input("Enter a two digit hex number: ")
2
3 def hexToDec(hex):
4     if hex == "0":
5         return 0
6     elif hex == "1":
7         return 1
8     elif hex == "2":
9         return 2
10    elif hex == "3":
11        return 3
12    elif hex == "4":
13        return 4
14    elif hex == "5":
15        return 5
16    elif hex == "6":
17        return 6
18    elif hex == "7":
19        return 7
20    elif hex == "8":
21        return 8
22    elif hex == "9":
23        return 9
24    elif hex == "A":
25        return 10
26    elif hex == "B":
27        return 11
28    elif hex == "C":
29        return 12
30    elif hex == "D":
31        return 13
32    elif hex == "E":
33        return 14
34    elif hex == "F":
35        return 15
36
37 decimal = hexToDec(n[0]) * 16 + hexToDec(n[1])
38 binary = ""
39 print(f"Decimal of {n} is {decimal}")
40 while decimal > 0:
41     binary = str(decimal % 2) + binary
42     decimal = decimal // 2
43 print(f"Binary of {n} is {binary}")
```

- Assignment2 (): python3 17_hexToBinDec.py
Enter a two digit hex number: AF
Decimal of AF is 175
Binary of AF is 10101111
- Assignment2 (): python3 17_hexToBinDec.py
Enter a two digit hex number: FA
Decimal of FA is 250
Binary of FA is 11111010
- Assignment2 (): python3 17_hexToBinDec.py
Enter a two digit hex number: FF
Decimal of FF is 255
Binary of FF is 11111111
- Assignment2 (): python3 17_hexToBinDec.py
Enter a two digit hex number: AA
Decimal of AA is 170
Binary of AA is 10101010
- Assignment2 ():

8. Swap Case

```
18_swapCase.py X
Elective > Assignment2 > 18_swapCase.py > ...
1 n = input("Enter a string: ")
2
3 print("String after swapping case: ", end="")
4
5 for c in n:
6     if c.islower():
7         print(c.upper(), end="")
8     elif c.isupper():
9         print(c.lower(), end="")
10    else:
11        print(c, end="")
12
13 print()
```

• Assignment2 (): python3 18_swapCase.py
Enter a string: My Name is Mayon
String after swapping case: mY nAME IS mAYON
○ Assignment2 ():

9. Encrypt a string using the shift cipher(key=3 Ceaser cipher)

```
19_ceaserCypher.py X
Elective > Assignment2 > 19_ceaserCypher.py > ...
1 inpStr = input("Enter a string: ")
2 inpKey = int(input("Enter a key: "))
3 outStr = ""
4
5 for c in inpStr:
6     if(c.isupper()):
7         outStr += chr((ord(c) + inpKey - 65) % 26 + 65)
8     elif(c.islower()):
9         outStr += chr((ord(c) + inpKey - 97) % 26 + 97)
10    else:
11        print("Only Alphabets are allowed")
12        exit()
13
14 print("CipherText:" + outStr)
15
16 # Decrypt above ceaser Cypher
17 print("Decrypted Text: ", end="")
18 for c in outStr:
19     if(c.isupper()):
20         print(chr((ord(c) - inpKey - 65) % 26 + 65), end="")
21     else:
22         print(chr((ord(c) - inpKey - 97) % 26 + 97), end="")
23
24 print("\n")
25
```

• Assignment2 (): python3 19_ceaserCypher.py
Enter a string: SuperSecret
Enter a key: 3
CipherText:VxshuVhfuhw
Decrypted Text: SuperSecret
○ Assignment2 ():

10. Check password strength

```
20_passwordStrength.py X
Elective > Assignment2 > 20_passwordStrength.py > ...
1 import getpass
2
3 password = getpass.getpass('Password: ')
4
5 # 1. Length
6 if(len(password) < 8):
7     print("Password is too short")
8     exit()
9
10 # 2. Upper Case
11 if(password.isupper()):
12     print("Password must contain lower case characters")
13     exit()
14
15 # 3. Lower Case
16 if(password.islower()):
17     print("Password must contain upper case characters")
18     exit()
19
20 # 4. Special Characters
21 if(password.isalnum()):
22     print("Password must contain special characters")
23     exit()
24
25 # 5. Numbers
26 if(password.isalpha()):
27     print("Password must contain numbers")
28     exit()
29
30 print("Password is strong")
31
```

• Assignment2 (): python3 20_passwordStrength.py
Password:
Password is too short
• Assignment2 (): python3 20_passwordStrength.py
Password:
Password must contain upper case characters
• Assignment2 (): python3 20_passwordStrength.py
Password:
Password is strong
○ Assignment2 ():