

Assignment 2: Programming in Python CST 362

Name: **Alby Thekkedan** Roll No **8** Class **CS6B**

KTU-ID: **MDL20CS129**

Date of submission: 06-March-2023

Learning outcome: Learn to use loops, nested loops and strings

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

```
import math
n=int(input("Enter a range "))
for i in range(0,n):
    print('(x^', (2*i+1)*pow(-1,i), ')/((', (2*i+1), ')!', '+', end=" ")
print("\nor")
for i in range(0,n):
    print('(x^', (2*i+1)*pow(-1,i), ')/((', math.factorial(2*i+1), ')', '+', end=" ")
```

Enter a range 5
 $(x^1)/(1!) + (x^{-3})/(3!) + (x^5)/(5!) + (x^{-7})/(7!) + (x^9)/(9!) +$
or
 $(x^1)/(1) + (x^{-3})/(6) + (x^5)/(120) + (x^{-7})/(5040) + (x^9)/(362880) +$

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

```
import math
n = int(input("Enter n: "))
x = float(input("Enter the value of x ")) #30 degree =0.5236 radian
sinx = x
sign = -1
for i in range(1, n):
    term = (x**(2*i+1)) / math.factorial(2*i+1)
    sinx = sinx + (sign * term)
    sign = sign*(-1)
print("sin(x) = ", sinx)
```

Enter n: 60

Enter the value of x 0
sin(x) = 0.0

Enter n: 9

Enter the value of x .5236
sin(x) = 0.500001060362603

3.

1 The pyramid is given for n=4 do this for any n

1 2

1 2 3

1 2 3 4

```
n=int(input("Enter the number "))
for i in range(n):
    for j in range(i+1):
        print(j+1,end=" ")
    print("\n")
```

Enter the number 4

1

1 2

1 2 3

1 2 3 4

4. Reverse a number and also find the sum of the digits Eg: i/p : 546

o/p: reverse=645 sum=15

Enter a number: 645546

The reverse is: 645546

The sum is: 30

Enter a number: 546

The reverse is: 645

The sum is: 15

```
num = int(input("Enter a number: "))
rev = 0
while num > 0:
    digit = num % 10
    rev = rev * 10 + digit
    num = num // 10
print("The reverse is: ", rev)
sum1 = 0
while rev > 0:
    digit = rev % 10
    sum1 += digit
    rev = rev // 10
print("The sum is: ", sum1)
```

Enter a number: 153

153 is an Armstrong number

Enter a number: 1634

1634 is an Armstrong number

Enter a number: 100

100 is not an Armstrong number

5. Armstrong numbers

```
num = int(input("Enter a number: "))
n = len(str(num))
sum1 = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum1 += digit ** n
    temp //= 10
if num == sum1:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

6. Find the square root of a number using Newton's method

Enter a number: 45
The square root of 45.0 is 6.708203932499369

Enter a number: 81
The square root of 81.0 is 9.0

```
num = float(input("Enter a number: "))
guess = num / 2 # Initial guess is half of the number

while True:
    new_guess = (guess + num / guess) / 2 # Compute the new guess
    if abs(new_guess - guess) < 1e-6: # Check for convergence
        break
    guess = new_guess # Update the guess

print("The square root of", num, "is", new_guess)
```

7. Write a program that computes an investment report.

```
Enter the investment amount: 10000.00
Enter the number of years: 5
Enter the rate as a %: 5
```

Year	Starting balance	Interest	Ending balance
1	10000.00	500.00	10500.00
2	10500.00	525.00	11025.00
3	11025.00	551.25	11576.25
4	11576.25	578.81	12155.06
5	12155.06	607.75	12762.82

```
Ending balance: $12762.82
Total interest earned: $2762.82
```

```
Enter the investment amount: 100000
Enter the number of years: 3
Enter the rate in percentage : 2
```

Year	Starting balance	Interest	Ending balance
1	100000.00	2000.00	102000.00
2	102000.00	2040.00	104040.00
3	104040.00	2080.80	106120.80

```
Ending balance: $106120.80
Total interest earned: $6120.80
```

```
first_Balance = float(input("Enter the investment amount: "))
years = int(input("Enter the number of years: "))
rate = int(input("Enter the rate in percentage : "))
rate = rate / 100
totalInterest = 0.0
print("%4s%18s%10s%16s" % ("Year", "Starting balance", "Interest", "Ending balance"))

for year in range(1, years + 1):
    interest = first_Balance * rate
    endBalance = first_Balance + interest
    print("%4d%18.2f%10.2f%16.2f" % (year, first_Balance, interest, endBalance))
    first_Balance = endBalance
    totalInterest += interest

print("Ending balance: $%.2f" % endBalance)
print("Total interest earned: $%.2f" % totalInterest)
```

8. Check whether the given number is a Krishnamurti number (.Use factorial () function from math)For example: 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145 is a Krishnamurthy Number

```
num = int(input("Enter the Number "))
temp = num
add = 0
while temp > 0:
    fact = 1
    i = 1
    rem = temp % 10
    fact = math.factorial(rem)
    add = add + fact
    temp = temp // 10

if add == num:
    print("\n%d is a Krishnamurthy Number." % num)
else:
    print("%d is Not a Krishnamurthy Number." % num)
```

Enter the Number 145

145 is a Krishnamurthy Number.

Enter the Number 156

156 is Not a Krishnamurthy Number.

9. Find the sum of the first and last digit of a number(i/p:354 o/p=3+4=7)

Enter a number: 354

The sum of the first and last digits of 354 is 7

Enter a number: 7854

The sum of the first and last digits of 7854 is 11

```
num = int(input("Enter a number: "))
first_digit = num
while first_digit >= 10:
    first_digit //= 10

last_digit = num % 10
sum_digits = first_digit + last_digit

print("The sum of the first and last digits of", num, "is", sum_digits)
```

10. Input a number and print it in words (i/p:345 o/p: Three Four Five)

```
num = int(input("Enter a number: "))
digits = ["zero", "one", "two", "three", "four",
          "five", "six", "seven", "eight", "nine"]

for d_char in str(num):
    d_int = int(d_char)
    dw = digits[d_int]
    print(dw, end=" ")
```

Enter a number: 84

eight four

Enter a number: 9845

nine eight four five

Strings

Outcome: Learn String Indexing and slicing, programming with strings

Consider the string str="Python Programming by Alby".Write statements in python to implement the following

```
string="Python Programming by Alby"
```

a) To display the last four characters.

```
print("Last 4 letters of of string >", string, "< are ", string[-4:])
```

b) To display the substring starting from index 4 and ending at index 8.

```
print("index from 4 to index 8 is > ",string[4:8]," <")
```

#c) Find the Length of the string,min and max(characters)

```
print("Length of string is ",len(string))
```

```
words = string.split() # Split the string into a list of words
```

```
min_length = len(min(words, key=len))
```

```
max_length = len(max(words, key=len))
```

```
print("Length of the shortest word:", min_length)
```

```
print("Length of the longest word:", max_length)
```

#d) —»To trim the last four characters from the string.

```
print("Trimmed the last four characters >",string[:-4]," <")
```

#e) To trim the first four characters from the string *****

```
print("Trimmed the first four characters >",string[4:]," <")
```

#f)»To display the starting index of the substring 'gr'.

```
print("Index of > gr < is at ",string.find("gr"))
```

#g)»To change the case of the given string small letter to capital & vice versa

```
print("Changes case ",string.swapcase())
```


Last 4 letters of of string > Python Programming by Alby < are Alby
 index from 4 to index 8 is > on P <
 Length of string is 26
 Length of the shortest word: 2
 Length of the longest word: 11
 Trimmed the last four characters > Python Programming by <
 Trimmed the first four characters > on Programming by Alby <
 Index of > gr < is at 10
 Changes case pYTHON pROGRAMMING BY aLBY

```
#h)*To check if the string is in title case
if string.istitle(): print(string, "is in title case.")
else: print(string, "is not in title case.")

#i) To replace all the occurrences of letter 'm' in the string with '*'
print("Changes case ",string.replace('m', '*'))

#j) Reverse of a string
print("Reverse is ",string[::-1])

#k) count all occurances of m
print("m is found ",string.count('m')," times")

#l)*characters in even positions 0,2,4,...
print("characters in even positions are ",string[::2])

#m)*characters in even positions 0,2,4,... in the reverse of string
print("characters in even positions in reverse are ",string[::-2])

#n) Check if the substring 'on' is in the string
if 'on' in string: print("'on' is present in", string)
else: print("'on' is not present in", string)

#o) First occurrence of t
print("1st occurrence of > t < is at ",string.find('t'))

#p) Convert string to upper
print("String in Upper case is ",string.upper())
```

Python Programming by Alby is not in title case.
 Changes case Python Progra**ing by Alby
 Reverse is yblA yb gnimmargorP nohtyP
 m is found 2 times
 characters in even positions are Pto rgamn yAb
 characters in even positions in reverse are yl bgimroPnh
 'on' is present in Python Programming by Alby
 1st occurrence of > t < is at 2
 String in Upper case is PYTHON PROGRAMMING BY ALBY

2. Write a program to check whether the given string is palindrome or not.

```
s=input("Enter a string ")
if s==s[::-1]:
    print("Palindrome")
else:
    print("Not palindrome")
```

Enter a stringmalayalam
 Palindrome
 Enter a string football
 Not palindrome

3. Count the vowels, digits, consonents, spaces in a string.

```
Enter a String CSBmec2024
Vowel count 1
Consonent count 5
Digit count 4
```

```
vow="aeiouAEIOU"
number="0123456789"
s=input("Enter a String ")
vc=0;dg=0;cc=0
for c in s:
    if c in vow:
        vc=vc+1
    if c not in vow:
        if c not in number:
            cc=cc+1
        if c in number:
            dg=dg+1

print("Vowel count ",vc)
print("Consonent count ",cc)
print("Digit count ",dg)
```

4. Read a binary number as a string and find its corresponding decimal

```
b=input("Enter a binary number as a string ")
l=len(b)
exp=l-1
dec=0
for n in b:
    dec+=int(n)*(2**exp)
    exp=exp-1
print(dec)
```

```
Enter a binary string 1010
10
Enter a binary number as a string 1101
13
```

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

```
dec=int(input("Enter a decimal "))
bin=""
while dec!=0:
    r=dec%2
    bin=str(r)+bin
    dec=dec//2
print(bin," Is its binary")
```

```
Enter a decimal 9
1001 Is its binary

Enter a decimal 4096
1000000000000 Is its binary
```

6. Read an 8 bit binary number and print the hex equivalent

```
Enter a binary number as a string 1101
Hexadecimal of 1101 is D
```

```
Enter a binary number as a string 1100010101101011
Hexadecimal of 1100010101101011 is C56B
```

```
b=input("Enter a binary number as a string ")
l=len(b)
exp=l-1
dec=0
for n in b:
    dec+=int(n)*(2**exp)
    exp=exp-1
hexa = ""
while dec > 0:
    rem = dec % 16
    if rem < 10:
        hexa = str(rem) + hexa
    else:
        hexa = chr(rem+55) + hexa
    dec//= 16
print("Hexadecimal of ", b, "is", hexa)
```

7. Read a two digit hex number and print the binary and decimal equivalent.

```
hex_num = input("Enter a two-digit hexadecimal number: ")
dec_num = 0
for digit in hex_num:
    dec_num = dec_num * 16 + int(digit,16)
print("Decimal equivalent:", dec_num)
bin_num = ''
while dec_num > 0:
    bin_num = str(dec_num % 2) + bin_num
    dec_num //= 2
print("Binary equivalent:", bin_num)
```

```
Enter a two-digit hexadecimal number: FF
Decimal equivalent: 255
Binary equivalent: 11111111

Enter a two-digit hexadecimal number: F0
Decimal equivalent: 240
Binary equivalent: 11110000
```

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

```
Enter a key 3
Enter a string xiyv
alby

Enter a key 3
Enter a string zootopia
crrwrsld
```

```
import string
alph=string.ascii_lowercase
k=int(input("Enter a key "))
ciphertext=""
plaintext=input('Enter a string ')
for c in plaintext:
    if c in alph:
        i=alph.find(c)
        i=(i+k)%26
        ciphertext=ciphertext+alph[i]
print(ciphertext)
```

9. Write a Python program to check the validity of a password given by the user. The Password should satisfy the following criteria: Contains at least one letter between a and z, number between 0 and 9, letter between A and Z, special character from \$, #, @ 5. Minimum length of password: 8

```
alph=string.ascii_lowercase
ALPHA=string.ascii_uppercase
numbers="0123456789"
spec_char="$@#"
```

```
def is_valid_password(password):
    if len(password) < 8:
        return False
```

```
    has_lowercase = False
    for char in password:
        if char in alph:
            has_lowercase = True
            break
    if not has_lowercase:
        return False
```

```
    has_uppercase = False
    for char in password:
        if char in ALPHA:
            has_uppercase = True
            break
    if not has_uppercase:
        return False
```

```
    has_digit = False
    for char in password:
        if char in numbers:
            has_digit = True
            break
    if not has_digit:
        return False
```

```
    has_special_char = False
    for char in password:
        if char in spec_char:
            has_special_char = True
            break
    if not has_special_char:
        return False

    return True
```

```
password = input("Enter a password: ")
if is_valid_password(password):
    print("Password is valid \n\U0001f600")
else:
    print("Invalid Password \n\nThe Password should satisfy the following criteria
```

Enter a password: 123344433#@@@@
Invalid Password

Enter a password: @mecEXCEL2024
Password is valid



Enter a password: davidbeckam
Invalid Password

The Password should satisfy the following criteria:
1. Contains at least one letter between a and z
2. Contains at least one number between 0 and 9
3. Contains at least one letter between A and Z
4. Contains at least one special character from \$, #, @
5. Minimum length of password: 8

Enter a password: lokesh3456L
Invalid Password

The Password should satisfy the following criteria:
1. Contains at least one letter between a and z
2. Contains at least one number between 0 and 9
3. Contains at least one letter between A and Z
4. Contains at least one special character from \$, #, @
5. Minimum length of password: 8