

# Programming in Python CST 362

## Assignment 6

Mayon Francis  
CS6A  
44

1. Write a Python script to find the sum of digits of a number adding repeatedly to single digit. Eg: 197 = 1+9+7=17=1+7=8.

```
1_digitSum.py U x
Assignment6 > 1_digitSum.py > ...
1 num = int(input("Enter a number: "))
2 sum = num%9
3 if sum == 0:
4     sum = 9
5
6 print("The sum of digits is", sum)
```

```
Assignment6 (main): python3 1_digitSum.py
Enter a number: 123
The sum of digits is 6
Assignment6 (main):
```

2. Write an efficient Python program to check whether the given number is power of 2.( Use bitwise operator)

```
2_power2.py U x
Assignment6 > 2_power2.py > ...
1 num = int(input("Enter a number: "))
2
3 if num&num-1 == 0:
4     print("The number is a power of 2")
5 else:
6     print("The number is not a power of 2")
```

```
Assignment6 (main): python3 2_power2.py
Enter a number: 4
The number is a power of 2
Assignment6 (main): python3 2_power2.py
Enter a number: 7
The number is not a power of 2
Assignment6 (main):
```

3. Let s="Python Programming" be a string. Write commands for the following i)reverse the string ii)print last 4 characters iii)count of 'P'

```
2_power2.py U 3_stringFunc.py U x
Assignment6 > 3_stringFunc.py > ...
1 s = "Python Programming"
2
3 print(s[::-1])
4 print(s[-4:])
5 print(s.count("P"))
```

```
Assignment6 (main): python3 3_stringFunc.py
gnimmargorP nohtyP
ming
2
Assignment6 (main):
```

4. Write lambda functions for the following i)F(x)=x\*\*3+3\*x +2 ii)Filter even numbers from a list

```
4_lambda.py U x
Assignment6 > 4_lambda.py > ...
1 f = lambda x: x**3 + 3*x + 2
2 print(f(2))
3
4 l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
5 l2 = list(filter(lambda x: x%2 == 0, l))
6 print(l2)
```

```
Assignment6 (main): python3 4_lambda.py
16
[2, 4, 6, 8, 10]
Assignment6 (main):
```

5. Write a Python program to find the largest digit in a number and also print the position ( assume non repeating digits)

```
5_largestDigit.py ×
Assignment6 > 5_largestDigit.py > ...
1 num = input("Enter a number: ")
2 temp = int(num)
3 largest = 0
4 while temp > 0:
5     digit = temp%10
6     if digit > largest:
7         largest = digit
8     temp = temp//10
9
10 index = num.index(str(largest)) + 1
11 print(f"The largest digit is {largest} at position {index}")

● Assignment6 (main): python3 5_largestDigit.py
Enter a number: 12346211
The largest digit is 6 at position 5
○ Assignment6 (main):
```

6. Write a Python program to read a binary number and find the equivalent decimal number. Consider binary point also.

```
6_binToDec.py ×
Assignment6 > 6_binToDec.py > ...
1 binStr = input("Enter a binary number: ")
2
3 integerPart = binStr.split(".")[0]
4 fractionalPart = binStr.split(".")[1]
5 decStr = 0
6
7 for idx, c in enumerate(integerPart):
8     decStr += float(c) * 2**(len(integerPart) - idx - 1)
9
10 for idx, c in enumerate(fractionalPart):
11     decStr += float(c) * 2**(-idx - 1)
12
13 print(decStr)
14

● Assignment6 (main): python3 6_binToDec.py
Enter a binary number: 1010.0101
10.3125
○ Assignment6 (main):
```

7. Write a Python program to find the value of  $\cos(x)$  up to  $n$  terms using the series  $1 - x^2/2! + x^4/4! - x^6/6! + \dots$ . Use your own factorial function.

```
7_cosSum.py ×
Assignment6 > 7_cosSum.py > ...
1 import math
2 n = int(input("Enter n: "))
3 x = int(input("Enter x: "))
4 sum=0
5 print("Series: ", end=" ")
6 for i in range(0, n+1, 2):
7     temp = 1
8     if(i == 1):
9         pass
10    elif i % 4 == 1:
11        print("+", end=" ")
12    else:
13        temp *= -1
14        print("-", end=" ")
15    if i == 1:
16        print("x", end=" ")
17    else:
18        print("x^{}/{}!".format(i,i), end=" ")
19    temp *= (x**i)/math.factorial(i)
20    sum += temp
21 print()
22 print("Sum: {}\n".format(sum))

● Assignment6 (main): python3 7_cosSum.py
Enter n: 4
Enter x: 2
Series: - x^0/0! - x^2/2! - x^4/4!
Sum: -3.6666666666666665
○ Assignment6 (main):
```

8. Write a Python program to check the validity of a password given by the user. 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: 6

```

8_password.py x
Assignment6 > 8_password.py > ...
1 import getpass
2
3 password = getpass.getpass('Password: ')
4 # 1. Length
5 if(len(password) < 6):
6     print("Password is too short")
7     exit()
8 # 2. Upper Case
9 if(password.isupper()):
10    print("Password must contain lower case characters")
11    exit()
12 # 3. Lower Case
13 if(password.islower()):
14    print("Password must contain upper case characters")
15    exit()
16 # 4. Special Characters
17 if(password.isalnum()):
18    print("Password must contain special characters")
19    exit()
20 # 5. Numbers
21 if(password.isalpha()):
22    print("Password must contain numbers")
23    exit()
24 print("Password is strong")
25
Assignment6 (main): python3 8_password.py
Password:
Password is strong
Assignment6 (main):

```

9. Write a recursive function in Python to generate n<sup>th</sup> Fibonacci number. Use this function to print the Fibonacci series.

```

9_fibonacci.py x
Assignment6 > 9_fibonacci.py > ...
1 def fib(n):
2     if n == 1:
3         return 0
4     elif n == 2:
5         return 1
6     else:
7         return fib(n-2) + fib(n-1)
8
9 for i in range(1, 10):
10    print(fib(i))
Assignment6 (main): python3 9_fibonacci.py
0
1
1
2
3
5
8
13
21
Assignment6 (main):

```

10. Read a text file “sample.txt” and encrypt this file using the shift cipher(Use suitable key).Save the encrypted file in “enc.txt”

```

10_caesarCypher.py x
Assignment6 > 10_caesarCypher.py > ...
5
6 for c in inpStr:
7     if(c.isupper()):
8         outStr += chr((ord(c) + inpKey - 65) % 26 + 65)
9     elif(c.islower()):
10        outStr += chr((ord(c) + inpKey - 97) % 26 + 97)
11    else:
12        print("Only Alphabets are allowed")
13        exit()
14
15 print("CipherText:" + outStr)
16 f2 = open("enc.txt", "w")
17 f2.write(outStr)
18
19 # Decrypt above ceaser Cypher
20 print("Decrypted Text: ", end="")
21 for c in outStr:
22     if(c.isupper()):
23         print(chr((ord(c) - inpKey - 65) % 26 + 65), end="")
24     else:
25         print(chr((ord(c) - inpKey - 97) % 26 + 97), end="")
26
27 print("\n")
Assignment6 (main): python3 10_caesarCypher.py
CipherText:Mjqqt
Decrypted Text: Hello
Assignment6 (main):

```