**Applied Coding - Language Skill (Using String)**

**Problem 1: Mystery Length Create a function removeMysteryLength that takes a string of alpha-numeric characters with the length appended to the string, removes the length part of it, returns the remaining string. For example, if input string is “JamesBond00712” where, 12 is length of string JamesBond007 then, return “JamesBond007”. Write a main program to demonstrate your function.**

#pythoncode:

def removeMysteryLength():

s=str(input())

l=len(s)

output=''

for i in range(0,l-len(str(l))):

output+=s[i]

return output

if \_\_name\_\_=="\_\_main\_\_":

print(removeMysteryLength())

#output1:

tharunsheshadri15

tharunsheshadri

#output2:

jamesbond00712

jamesbond007

**Problem 2: Palindrome Check A string is a palindrome if it is identical forward and backward. For example “anna”, “civic”, “level” and “hannah” are all examples of palindromic words. Create a function named isPalindrome that takes a string from the user and determines whether or not it is a palindrome. Write a main program to demonstrate your function. Problem 2: Palindrome Check A string is a palindrome if it is identical forward and backward. For example “anna”, “civic”, “level” and “hannah” are all examples of palindromic words. Create a function named isPalindrome that takes a string from the user and determines whether or not it is a palindrome. Write a main program to demonstrate your function.**

1)using while loop:

def isPalindrome():

i=len(s)-1

output=''

while(i>=0):

output+=s[i]

i=i-1

if(output==s):

return True

else:

return False

if \_\_name\_\_=='\_\_main\_\_':

s=input()

print(isPalindrome())

#output: #output2;

Tharun reviver

False True

2)using slice operator:

def isPalindrome():

output=s[::-1]

if(output==s):

return True

else:

return False

if \_\_name\_\_=='\_\_main\_\_':

s=input()

print(isPalindrome())

#output: #output2:

hannah Algorithmica

True False

3)Using reversed keyword:

def isPalindrome():

r=reversed(s)

output=''.join(r)

if(output==s):

return True

else:

return False

if \_\_name\_\_=='\_\_main\_\_':

s=input()

print(isPalindrome())

#output: #output2

Madam madam

False True

4)Using for loop:

def isPalindrome():

output=''

for i in s:

output=i+output

if(output==s):

return True

else:

return False

if \_\_name\_\_=='\_\_main\_\_':

s=input()

print(isPalindrome())

#output:

Tharun level

False True

**Problem 3: Reverse Words Create a function named reverseWords that takes a string from the user and returns a string with words reversed. For example, if input string is “India is Best” then the returned string is “**Best **is India” Write a main program to demonstrate your function.**

#PYTHONCODE

def reverseWords():

l=s.split()

print(l)

l1=l[::-1]

output=' '.join(l1)

print(output)

if \_\_name\_\_=='\_\_main\_\_':

s=input()

reverseWords()

#output1

i love algorithmicA

['i', 'love', 'algorithmicA']

algorithmicA love i

#output2:

thinking with logic is called coding

['thinking', 'with', 'logic', 'is', 'called', 'coding']

coding called is logic with thinking

**Problem 4: Encode & Decode Create a function decode that takes a character as input and returns the ASCII code of that character. Create a function encode that takes a ASCII code(number) as input and returns the character associated with it. Write a main program to demonstrate your function.**

#pythoncode:

def Encode():

print(ord(c))

def decode():

print(chr(n))

if \_\_name\_\_=='\_\_main\_\_':

c=input()

n=int(input())

Encode()

decode()

#output:

c

103

99

g

**Problem 5: Caesar Cipher One of the first known examples of encryption was used by Julius Caesar. The idea behind Caesar cipher is simple. It starts by taking shift amount and message as input, then each letter in the original message is shifted(with wrap around) by shift amount places. For example, if shift amount=3 and message=axy then the result must be dab. Create a function named encrpytWithCaesar that takes a string, shift amount as input and returns the encrypted string as suggested by Caesar cipher. Ensure that your program encodes both uppercase and lowercase letters. Non-letter characters are not modified by the cipher. Write a main program to demonstrate your function.**

#pythoncode:

def encrpytWithCaesar(plaintext,n):

output = ""

for i in range(len(plaintext)):

ch = plaintext[i]

if ch==" ":

output+=" "

elif (ch.isupper()):

output += chr((ord(ch) + n-65) % 26 + 65)

elif(ch.islower()):

output += chr((ord(ch) + n-97) % 26 + 97)

return output

if \_\_name\_\_=="\_\_main\_\_":

plaintext = input()

n = int(input())

print("Cipher Text : " + encrpytWithCaesar(plaintext,n))

#output1:

tharun

3

Cipher Text : wkduxq

#output2:

ALGORITHMICA

6

Cipher Text : GRMUXOZNSOIG

**Problem 6: Does the string represent valid Integer? Write a function named isInteger that determines whether or not the characters in a string represent a valid integer. When determining if a string represents an integer you should ignore any leading or trailing white space. Once this white space is ignored, a string represents an integer if its length is at least 1 and it only contains digits, or if its first character is either + or - and the first character is followed by one or more characters, all of which are digits. Write a main program that reads a string from the user and reports whether or not it represents an integer.**

#pythoncode:

a=input()

a=a.strip()

if len(a)<1:

print("enter valid input")

else:

if all(a[0] in "0123456789" for i in range(len(a))):

print("it is valid integer")

elif (a[0] in "+-") and \

all(a[0] in "1234" for i in range(1,len(a))):

print('it is valid integer')

else:

print("invalid integer")

output:

674975476576

it is valid integer

#output:

kjhnvidhvyhvyu

invalid integer