**3/2/2019**

**2/02 Assignments Revision**

A

B A B

C B A B C

D C B A B C D

=>same logic of number pyramid with print (chr(x+64))

**function with default argument**

**def(a,b=10)**

**def add(a,b=10,c) #error**

**# def add(10) => a=10,b=10**

**# def add(10,20,30) => a=10,b=20,c=30**

**#WAP to perform addition and multiplication of given numbers(Input upto 5 numbers)**

def multiply(a,b=1,c=1,d=1,e=1):

return a\*b\*c\*d\*e

def add(a,b,c=0,d=0,e=0)

return a+b+c+d+e

def main():

print multiply(2,4)

print multiply(2,4,5)

print multiply(2,4,5,6)

print multiply(2,4,5,6,7)

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

main()

**# Variable number of Arguments in function**

**def Add(\*args):**

**print(type(args))**

**print(args)**

**#Program to add n number of arguments**

def Add(\*args):

print(type(args))

print(args)

result=0

for x in args:

result+=x

return result

def main():

print(Add(1,2))

print(Add(100,200,300))

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

main()

**#Variable arguments Dictionary(Key=>Value pair)**

def VariableArgsDictionary(a,b,\*args,\*\*kwargs):

print(a,b)

print(type(args),type(kwargs))

for x in args:

print(x)

for x in kwargs:

print(x,kwargs[x])

def main():

VariableArgsDictionary(10,20,1,2,3,4,5,6,7,name="Swapnil",hobby="Driving")

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

main()

**# WAP to accept number and check if it is even or odd without using arithmetic operator**

def IsOdd(no): #check value of LSB if it is 0 then number is odd if 1 number is odd

if((no&1)==0):

return false

else:

return true

**#WAP to accept string from user and print count of consonants from it.**

# count of consonants

def CountConsonants(inputString):

count=0

for x in inputString:

if (x not in ('aeiouAEIOU')):

count+=1

return count

def main():

inputString = input("Enter any String")

print(CountConsonants(inputString))

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

main()

**#WAP to accept number from user and bit position to turn off from the given number. Print number in decimal after turning off.**

**e.g. input 16, and bit position 5, LSB=1 Output=>0**

**logic : 16= 00010000**

**& 11101111**

**-----------------**

**00000000**

**STEP 1: 00000001 take x=1**

**STEP2: 00010000 shift 1 to position which is to turn off x=x<<(pos-1)**

**STEP 3: 11101111 negate the number x=-x & perform & with input number**

def turnOff(no,pos):

inverted=0

x=1

x=x<<(pos-1)

x=-x

inverted=(no & x)

return inverted

def main():

no=input("Enter number:")

pos=input("Enter Position:")

print(turnOff(no,pos))

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

main()

**#to turn off given number of bits with given number of positions (shift with compliment)**

**Homework**

**# Draw mindmap diagram for functions**

**#WAP to accept number and check if it is divisible by 8 without using arithmetic operators (hint: write binary of divisible by 8 and check if follows same pattern,if following it is divide by 8 else not)**

**#WAP to acccpet number and number of bits to turn ON from given position.(hint: use OR and just shift without compliment)**

**#WAP to accept string which has repetitive characters consecutively and optimize storage space for the same. e.g.input : aaabbccccdddddaa=>Output : a3b2c4d5a2**

**#draw pattern :**

**\*\*\*\*\*\*\***

**\*\*\* \*\*\***

**\*\* \*\***

**\* \***

**\*\* \*\***

**\*\*\* \*\*\***

**\*\*\*\*\*\*\***