**Exercise No:** 1

**Date:** 10.10.2020

**Aim:**

To write a Python program to print the calendar of a given month and year.

**Program:**

import calendar

y = int(input())

m = int(input())

if(y>999 and y<10000 and m>0 and m <13):

print(calendar.month(y, m))

else:

print("Invalid Input")

**Link:**

[**http://103.53.53.18/mod/vpl/forms/edit.php?id=229&userid=1764#**](http://103.53.53.18/mod/vpl/forms/edit.php?id=229&userid=1764)

**Output:**

2020

11

November 2020

Mo Tu We Th Fr Sa Su

1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30   
**Result:**

Calendar of the given year and month is obtained by calender import function.

**Exercise No:** 2

**Date:** 10.10.2020

**Aim:**

To write a python program with a list of a number x, count number of occurrences of x in the given list. Using a function **countX(lst, x)** to count the number x in a given list of numbers

**Program:**

def countX(lst,x):

count = 0

for elem in lst:

if elem == x:

count = count+1

print(count)

lst=[]

n=int(input())

for i in range(0,n):

elem = input()

lst.append(elem)

x=input()

countX(lst,x)

**Link:**

[**http://103.53.53.18/mod/vpl/forms/edit.php?id=230&userid=1764#**](http://103.53.53.18/mod/vpl/forms/edit.php?id=230&userid=1764)

**Output:**

6

1 1 1 1 3 5 6

1

4

**Result:**

The number x in the given list is obtained by using a function countX(lst, x)

**Exercise No:** 3

**Date:** 10.10.2020

**Aim:**

To write a Python program to remove and print every second number from a list of numbers until the list becomes empty.

**Program:**

int\_list=[]

n=int(input())

for j in range(0,n):

ele = input()

int\_list.append(ele)

def removeThirdNumber(int\_list):

pos = 2-1

i=0

len\_list = (len(int\_list))

while(len\_list > 0):

i = (pos + i) % len\_list

print(int\_list.pop(i))

len\_list-=1

removeThirdNumber(int\_list)

**Link:**

[**http://103.53.53.18/mod/vpl/forms/edit.php?id=231&userid=1764#**](http://103.53.53.18/mod/vpl/forms/edit.php?id=231&userid=1764)

**Output:**

5

1 2 3 4 5

2 4 1 5 3

**Result:**

Every second number from a list of numbers is printed until the list becomes empty and the output is obtained.

**Exercise No:** 4

**Date:** 11.10.2020

**Aim:**

To write a Python program (function) to print a single string from two set of strings received from user and swap the first two characters of each string

**Program:**

def swap(s):

new=""

for x in s:

new+=x

return new

a=input()

b=input()

s1=a.strip()

s2=b.strip()

if(len(s1)>=2 and len(s2)>=2):

t1=list(s1)

t2=list(s2)

c1=t1[0]

c2=t1[1]

t1[0]=t2[0]

t1[1]=t2[1]

t2[0]=c1

t2[1]=c2

s1=swap(t1)

s2=swap(t2)

print(s1," ",s2)

else:

print("Invalid")

**Link:**

[**http://103.53.53.18/mod/vpl/forms/edit.php?id=234&userid=1764#**](http://103.53.53.18/mod/vpl/forms/edit.php?id=234&userid=1764)

**Output:**

Tea

Coffee

Coa teffee

**Result:**

The two set of strings received from user and swap the first two characters of each string is performed and output is obtained.

**Exercise No:** 5

**Date:** 11.10.2020

**Aim:**

To write a Python program to Print the alphabets that are missing in that string to make it a pangram.

**Program:**

a=input()

a=set(a.strip())

a.remove(chr(32))

b="abcdefghijklmnopqrstuvwxyz"

b=set(b)

b=b.difference(a)

b=list(b)

b.sort()

print("".join(b))

**Link:**

[**http://103.53.53.18/mod/vpl/forms/edit.php?id=235&userid=1764#**](http://103.53.53.18/mod/vpl/forms/edit.php?id=235&userid=1764)

**Output:**

He is playing

bcdfjkmoqrtuvwxz  **Result:**

The alphabets that are missing in that string to make it a pangram are printed.

**Exercise No:** 6

**Date:** 11.10.2020

**Aim:**

To predict the output for the given python program.

**Program:**

PREDICT THE OUTPUT:

# Create a tuple, also called tuple packing.

numbers = 1, 2

print(numbers)

(1, 2)|((1,2)

# Create tuple with paranthesis.

numbers = (1, 2, 3)

print(numbers)

(1, 2, 3)|(1,2,3)

# Create an empty tuple.

numbers = ()

print(numbers)

**()|( )**

# Create a tuple with one item. Note that the trailing comma is necessary

numbers = 1,

print(numbers)

1

# Create a tuple with heterogenous items.

random\_tuple = "Hey", (1, 2), 1, ["you"]

print(random\_tuple)

("Hey", (1, 2), 1, ["you"])|("Hey",(1,2),1,["you"])

# Create tuple with tuple() constructor.

numbers = tuple()

print(numbers)

()|( )

numbers = tuple([1, 2]) # Takes any sequence as input

print(numbers)

(1, 2)|(1,2)

#### Methods on tuples #####

# Get length of list by using len() method.

numbers = 5, 8, 8

print(len(numbers))

3

# Get index of an element using the index() method.

numbers = 5, 8, 8

print(numbers.index(8))

1

# Count occurences of an item in a tuple.

numbers = 5, 8, 8

print(numbers.count(8))

2

eggs = ('hello', 42, 0.5)

eggs[0]

'hello'

hello

eggs[1:3]

(42, 0.5)

len(eggs)

3

# Access elements of a tuple by indexing.

str\_tuple = "hey", "there!", "how", "are", "you?"

print(str\_tuple[0])

hey

print(str\_tuple[len(str\_tuple) - 1])

you?

print(str\_tuple[-1])

you?

# Slicing a tuple.

str\_tuple = "hey", "there!", "how", "are", "you?"

print(str\_tuple[2:])

('how', 'are', 'you?')

print(str\_tuple[:2])

('hey', 'there!')

print(str\_tuple[-3:])

('how', 'are', 'you?')

print(str\_tuple[:-3])

('hey', 'there!')

print(str\_tuple[1:4])

('there!', 'how', 'are')

# Get a copy of the tuple by slicing.

print(str\_tuple[:])

('hey', 'there!', 'how', 'are', 'you?')

# Concatenate tuples.

numbers = (1, 2)

strings = ("Hey", "there")

print(numbers + strings)

(5, 8, 8, 'Hey', 'there')

(1, 2, "Hey", "there")

# Looping through tuple using 'in'.

numbers = 1, 2

for number in numbers:

print(number)

1,2

1 2

# Check if element is present in tuple.

numbers = 1, 2

print(1 in numbers)

True

print(5 in numbers)

False

# Tuple packing.

# We are packing two items 1 and 2 into the tuple.

numbers = 1, 2

# Tuple sequence unpacking.

# Number of variables used has to be same as the number of items in the tuple.

# Unpacking the tuple and assigning its items to x and y.

x, y = numbers

# Note that this is also packing the args as a tuple which gets unpacked as the print method's arguments.

print(x, y)

1 2

**Link:**

[**http://103.53.53.18/mod/hvp/view.php?id=238**](http://103.53.53.18/mod/hvp/view.php?id=238)

**Result:**

The output for the given program is obtained.