**PYTHON LABORATORY**

**Exercise No:** 1

**Date:**11.10.2020

**Aim:**

To write a python program to print the calendar of the 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/submissionview.php?id=229&userid=1780>

**Output:**

2020

12

December 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 31

**Result:**

Using calendar import function, Calendar of the given year and month is obtained.

**Exercise No**:2

**Date:**11.10.2020

**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 ele in lst:

if ele == x:

count = count +1

print(count)

lst = []

n = int(input())

for i in range(0,n):

ele = int(input())

lst.append(ele)

x= int(input())

countX(lst,x)

**Link:**

<http://103.53.53.18/mod/vpl/forms/submissionview.php?id=230&userid=1780>

**Output:**

5

2 5 5 6 6

6

3

**Result:**

Using a function **countX(lst, x)** to count the number x in a given list of numbers is obtained.

**Exercise No:**3

**Date**:11.10.2020

**Aim:**

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

**Program:**

def removeThirdNumber(int\_list):

position = 2-1

index=0

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

while len\_list>0:

index= (position + index)%len\_list

print(int\_list.pop(index))

len\_list -=1

n=int(input())

int\_list=[]

for i in range(n):

int\_list.append(int(input()))

removeThirdNumber(int\_list)

**Link:**

<http://103.53.53.18/mod/vpl/forms/submissionview.php?id=231&userid=1780>

**Output:**

6

1 2 3 4 5 6

2 4 6 3 1 5

**Result:**

Using removeThirdNumber(int\_list) function we printed by removing every second numbers from the list of numbers.

**Exercise No:**4

**Date:** 12.10.2020

**Aim:**

To write a python program to print 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 i in s:

new+=i

return new

fst=input()

snd=input()

sfst=fst.strip()

ssnd=snd.strip()

if(len(sfst)>=2 and len(ssnd)>=2):

tempfst=list(sfst)

tempsnd=list(ssnd)

charfst=tempfst[0]

charsnd=tempfst[1]

tempfst[0]=tempsnd[0]

tempfst[1]=tempsnd[1]

tempsnd[0]=charfst

tempsnd[1]=charsnd

sfst=swap(tempfst)

ssnd=swap(tempsnd)

print(sfst," ",ssnd)

else:

print("Invalid")

**Link:**

<http://103.53.53.18/mod/vpl/forms/submissionview.php?id=234&userid=1780>

**Output:**

 Python

    Java

jathon pyva

**Result:**

Using function swap(s) we can get the single string by swapping first two letters from two strings.

**Exercise No:**5

**Date:**12.10.2020

**Aim:**

To write a Python function to print missing characters to make string 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/submissionview.php?id=235&userid=1780>

**Output:**

the quick brown fox jumps over the lazy

dg

**Result:**

Using python function we printed the missing characters of the alphabets from the string.

**Exercise No:**6

**Date:**12.10.2020

Predict the Output

# Create a tuple, also called tuple packing.  
numbers = 1, 2  
print(numbers)    
# Create tuple with paranthesis.  
numbers = (1, 2, 3)  
print(numbers)   
# Create an empty tuple.  
numbers = ()  
print(numbers)   
# Create a tuple with one item. Note that the trailing comma is necessary  
numbers = 1,  
print(numbers)   
# Create a tuple with heterogenous items.  
random\_tuple = "Hey", (1, 2), 1, ["you"]  
print(random\_tuple)   
# Create tuple with tuple() constructor.  
numbers = tuple()  
print(numbers)   
numbers = tuple([1, 2]) #  Takes any sequence as input  
print(numbers) 

Predict the Output

#### Methods on tuples #####  
# Get length of list by using len() method.  
numbers = 5, 8, 8  
print(len(numbers))   
# Get index of an element using the index() method.  
numbers = 5, 8, 8  
print(numbers.index(8))   
# Count occurences of an item in a tuple.  
numbers = 5, 8, 8  
print(numbers.count(8)) 

eggs = ('hello', 42, 0.5)   
eggs[0]   
eggs[1:3]   
len(eggs) 

Predict the Output

# Access elements of a tuple by indexing.  
str\_tuple = "hey", "there!", "how", "are", "you?"  
print(str\_tuple[0])   
print(str\_tuple[len(str\_tuple) - 1])   
print(str\_tuple[-1]) 

# Slicing a tuple.

str\_tuple = "hey", "there!", "how", "are", "you?"  
print(str\_tuple[2:])   
print(str\_tuple[:2])   
print(str\_tuple[-3:])   
print(str\_tuple[:-3])   
print(str\_tuple[1:4])   
# Get a copy of the tuple by slicing.  
print(str\_tuple[:]) 

Predict the Output

# Concatenate tuples.  
numbers = (1, 2)  
strings = ("Hey", "there")  
print(numbers + strings) 

# Looping through tuple using 'in'.  
numbers = 1, 2  
for number in numbers:  
  print(number)   
    
# Check if element is present in tuple.  
numbers = 1, 2  
print(1 in numbers)   
print(5 in numbers) 

# 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) 

By,

Jeevitha. R

1518102048 CSE-D