LABCYCLE 1

PageNo:1 **EXPERIMENT NO:1**

Date:10/11/2022 **PROGRAM NAME: LEAPYEAR**

AIM

Display future leap years from current year to a final year entered by user.

ALGORITHM

```
Step 1: Start.
Step 2: Input current year and future year.
Step 3: Repeat step 3 to 7 from current year <=future year.
Step 4: Check whether year%4==0, if true then move to next step. If false then
move to step 6.
Step 5: Check whether year%4==100, if true then move to next step. If false
then move to step 6.
Step 6: Check whether year%4==0, if true then move to next step. If false then
move to step 6.
Step 7: Print leap years.
Step 8: Stop.
```

SOURCE CODE

```
year=int(input("Enter the current year:\n"))
fut=int(input("Enter the future year:\n"))
print("The leap years are")
for year in range(year,fut+1):
    if(year%4==0) and year%100!=0 or year%400==0:
        print(year)
```

OUTPUT

```
Enter the current year:
2022
Enter the future year:
2050
The leap years are
2024
2028
2032
2036
2040
2044
2048
```

RESULT

Program to display leap entered by user has been executed successfully and the output is obtained.

PageNo:3 **EXPERIMENT NO:2**

Date:10/11/2022 **PROGRAM NAME: LIST OF VALUES**

AIM

List comprehensions:

(a) Generate positive list of numbers from a given list of integers .

(b) Square of N numbers .

(c) Form a list of vowels selected from a given word.

(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal
values).

(a)Generate positive list of numbers from a given list of integers.

ALGORITHM

```
Step 1: Start.
Step 2:Input a list of positive and negative integers.
Step 3: Repeat step 4 until end of list.
Step 4: If i>0 then append to new list.
[end of loop]
Step 5: Print the new list.
Step 6: Stop
```

SOURCE CODE

```
lista=[5,6,7,-7,-3,-5,10,-15,2,11]
li=[]
print("The positive numbers in the list are\n")
for i in lista:
   if i>0:
      li.append(i)
```

```
print(li)
```

OUTPUT

```
 The positive numbers in the list are

[5, 6, 7, 10, 2, 11]
```

RESULT

The program to print positive numbers from a given list has been executed successfully and the output is verified.

PageNo:5

Date:10/11/2022

PROGRAM NAME: SQUARE OF N NUMBERS

AIM

(b) Square of N numbers .

ALGORITHM

```
Step 1: Start.
Step 2: Input the number, n.
Step 3: Repeat step 4 with range (1, n+1).
Step 4: Find square, sq=i*i.
Print(sq).
Step 5: Stop.
```

SOURCE CODE

```
n=int(input("Enter the limit to find the square of N numbers:\n"))
for i in range(1,n+1):
sq=i*i
print("The squares of ",i,"is",sq)
```

OUTPUT

```
Enter the limit to find the square of N numbers:
5
The squares of 1 is 1
The squares of 2 is 4
The squares of 3 is 9
The squares of 4 is 16
The squares of 5 is 25
```

RESULT

The program executed successfully and the output is verified.

PageNo:6

Date:10/11/2022 **PROGRAM NAME: LIST VOWELS**

AIM

(C)Form a list of vowels selected from a given word.

ALGORITHM

```
Step 1: Input empty list.
Step 2: Input a word.
Step 3: check if each letter of word present in list of vowels, if true.
append the letter to empty list, goto step 4.
Step 4: Print list.
Step 5: Stop.
```

SOURCE CODE

```
l=[]
word=input("Enter a word:\t")
vowels=['a','e','i','o','u','A','E','I','O','U']
for i in word:
   if i in vowels:
      l.append(i)
print("The vowels present in the word are \t",l)
```

OUTPUT

```
Enter a word:    Encyclopedia
The vowels present in the word are    ['E', 'o', 'e', 'i', 'a']
```

RESULT

The program has been executed successfully and the output is verified.

PageNo:7

Date:10/11/2022 **PROGRAM NAME: ORDINAL VALUE**

AIM

(d) List ordinal value of each element of a word(Hint:use ord( ) to get ordinal values).

ALGORITHM

```
Step 1:Start.
Step 2:Input a word.
Step 3:Print the ordinal value by iterating the word.
```

SOURCE CODE

```
word=input("Enter word:")
print([ord(i) for i in word])
for i in word:
   print("Ordinal value of",i,"is:",ord(i))
```

OUTPUT

```
Enter word:college
[99, 111, 108, 108, 101, 103, 101]
Ordinal value of c is: 99
Ordinal value of o is: 111
Ordinal value of l is: 108
Ordinal value of l is: 108
Ordinal value of e is: 101
Ordinal value of g is: 103
Ordinal value of e is: 101
```

RESULT

The program has been executed successfully and the output is verified.

PageNo:8
Date:21/11/2022

EXPERIMENT NO:3
PROGRAM NAME: OCCURENCES OF WORD

AIM

Count the occurrences of each word in a line of text.

ALGORITHM

```
Step 1: Start.
Step 2: Set Define function word_count(str)
count=dict()
words=str.split()
Take the count of each word.
Print(count).
Step 3: Input a string.
Step 4: Call function, word_count(str).
Step 5: Stop.
```

SOURCE CODE

```
def word_count(str):
    counts=dict()
    words=str.split()
    for word in words:
        if word in counts:
            counts[word]+=1
        else:
            counts[word]=1
    return counts
wrd=str(input("Enter a sentence: "))
print(word_count(wrd))
```

OUTPUT

```
Enter a sentence: the rises in the east and sets in the west
{'the': 3, 'rises': 1, 'in': 2, 'east': 1, 'and': 1, 'sets': 1, 'west': 1}
```

RESULT

The program to count the occurrences of each word in a line has been executed successfully and the ouput is verified.

PageNo:10 **EXPERIMENT NO:4**

Date:21/11/2022 **PROGRAM NAME: LIST OF INTEGERS**

AIM

Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

ALGORITHM

Step 1: Start.
Step 2: Input the two empty list.
Step 3: Input the limit,n.
Step 4: Append each element to the list.
Step 5: Append each element greater than 100 in another list.
Step 6: Print new list.
Step 7: Stop.

SOURCE CODE

```
lis=[]
a=[]
n=int(input("Enter the limit of the list \t"))
print("Enter the list of elements")
for i in range(0,n):
print("Enter the element no:-{}:".format(i+1))
elm=int(input())
lis.append(elm)
print("The entered list is ",lis)
for i in lis:
if i>100:
a.append(i)
else:
print("Over")
print("The values greater than 100 ",a)
```

OUTPUT

```
Enter the limit of the list:   5
Enter the list of elements.
Enter the element no:1:
216
Enter the element no:2:
100
Enter the element no:3:
99
Enter the element no:4:
101
Enter the element no:5:
89
The entered list is  [216, 100, 99, 101, 89]
Over
Over
Over
The values greater than 100  [216, 101]
```

RESULT

The program to display list of integers has been executed successfully and the output is verified.

PageNo:12

Date:21/11/2022

EXPERIMENT NO:5
PROGRAM NAME: OCCURRENCE OF ‘a’

AIM

Store a list of first names.Count the occurrences of ‘a’ within the list.

ALGORITHM

```
Step 1: Start.
Step 2: Input an empty list and limit of list.
Step 3: Append each element to the list.
Step 4: Initialize count=0.
Step 5: Check the presence of “a” in the list. Update count.
Step 6: Print count.
Step 7: Stop.
```

SOURCE CODE

```
list1=[]
len=int(input("Enter the number of names you want to insert : "))
for i in range(0,len):
    print("Enter the name ",i+1," you want to insert ")
    fname=input()
    list1.append(fname)
    count_a=0
for names in list1:
    count_a+=names.count("a")
print("Occurrence of ‘a’ in given list is",count_a)
```

OUTPUT

```
Enter the number of names you want to insert:4
Enter the name  1  you want to insert
alex
Enter the name  2  you want to insert
neefa
Enter the name  3  you want to insert
alan
Enter the name  4  you want to insert
jeena
 Occurrence of 'a' in given list is 5
```

RESULT

The Program to count the occurrences of ‘a’ within the list has been executed successfully and output is verified.

PageNo:14 **EXPERIMENT NO:6**

Date:21/11/2022 **PROGRAM NAME:COMPARE TWO LISTS**

AIM

Enter 2 lists of integers. Check
(a) Whether lists are of same length.
(b) whether list sums to same value.
(c) whether any value occur in both.

ALGORITHM

```
Step 1: Start.
Step 2: Input two strings.
Step 3: Get the length of two strings. Compare the lengths and print.
Step 4: Find the sum of elements in the list and compare and print.
Step 5: Find the similar value occurrence and print.
Step 6: Stop.
```

SOURCE CODE

```
def length(flist,slist):
	print("a.Length of list 1:\t",len(flist))
	print("\tLength of list 2:\t",len(slist))
	if len(flist)==len(slist):
		print("\tBoth list have same size")
	else:
		print("Different length ")


def sumoflist(flist,slist):
	s1=0
	s2=0
	for num in flist:
        s1+=num
    for num in slist:
        s2+=num
    if s1==s2:
        print("b.Sum are same ",s1," ,",s2)
    else:
        print("b.Sum are different for both list ",s1," ,",s2)

def findele(flist,slist):
    for num in flist:
        if num in slist:
            print("c.",num," found in both list\n")
# driver code
flist=[]
slist=[]
len1=int(input("Enter the number of elements you want to add on list 1:
"))
for i in range(0,len1):
    print("Enter the element ",i+1)
    inp=int(input())
    flist.append(inp)
len2=int(input("Enter the number of elements you want to add on list 2
:"))
for i in range(0,len2):
    print("Enter the element ",i+1)
    inp=int(input())
    slist.append(inp)
length(flist,slist)
sumoflist(flist,slist)
findele(flist,slist)
```

OUTPUT

```
Enter the number of elements you want to add on list 1: 5
Enter the element  1
56
Enter the element  2
34
Enter the element  3
23
Enter the element  4
11
Enter the element  5
12
Enter the number of elements you want to add on list 2: 3
Enter the element  1
23
Enter the element  2
30
Enter the element  3
12
a.Length of list 1:        5
        Length of list 2:  3
Different length
b.Sum are different for both list  136  , 65
c. 23  found in both list

c. 12  found in both list
```

RESULT

Program to compare two list has been executed successfully and the output is verified.

PageNo:17 **EXPERIMENT NO:7**

Date:05/12/2022 **PROGRAM NAME:REPLACE CHARACTER**

AIM

Get a string from an input string where all occurrences of first character replaced with '$', except first character. [eg: onion -> oni$n]

ALGORITHM

```
Step 1: Stop.
Step 2: Input the string.
Step 3: Replace the character with '$' using replace().
Step 4: Print string.
Step 5: Stop.
```

SOURCE CODE

```
str1=input("Enter a string:")
for i in range (1,len(str1)):
   str2=str1[0]+str1[1:].replace(str1[0],'$')
print("String after replaced with '$' is ",str2)
```

OUTPUT

```
Enter a string:elephant
String after replaced with '$' is  el$phant.
```

RESULT

The program to replace character has been executed successfully and the output is verified.

PageNo:18 **EXPERIMENT NO:8**

Date:05/12/2022 **PROGRAM NAME :EXCHANGE CHARACTERS**

AIM

Create a string from given string where first and last characters exchanged. [eg: python -> nythop]

ALGORITHM

```
Step 1: Start.
Step 2: Input a string to a variable.
Step 3: Define a function,
Store last, first and middle character to a variable.
Print swapped string.
Step 4: Stop.
```

SOURCE CODE

```
def swap(string):
      # storing the first character
    start = string[0]
     # storing the last character
    end = string[-1]
    swapped_string = end + string[1:-1] + start
    print(swapped_string)
a=input("Enter the string: ")
swap(a)
```

OUTPUT

```
Enter the string Football:
lootbalF
```

RESULT

Program to swap first and last character of a string has been executed successfully and output is verified.

PageNo:19 **EXPERIMENT NO:9**

Date:05/12/2022 **PROGRAM NAME :AREA OF CIRCLE**

AIM

Accept the radius from user and find area of circle.

ALGORITHM

```
Step 1: Start.
Step 2: Input radius and pi value(3.147).
Step 3: Compute area=pi*r*r
Step 4: Print result.
Step 5: Stop
```

SOURCE CODE

```
def area_of_circle(r):
    pi=3.147
    area=pi*r*r
    return area
radius=float(input("Enter the radius: "))
print("Area of the circle for given radius is ",area_of_circle(radius))
```

OUTPUT

```
Enter the radius: 5
Area of the circle for given radius is  78.675
```

RESULT

The program to find area of circle has been executed successfully and the output is obtained.

PageNo:20 **EXPERIMENT NO:10**

Date:05/12/2022 **PROGRAM NAME :BIGGEST NUMBER**

AIM

Find biggest of 3 numbers entered.

ALGORITHM

```
Step 1: Start.
Step 2: Input 3 numbers.
Step 3: Check
          If a>b and a>c:
               then  Print(a)
          If b>a and b>c:
          then  Print(b)
          Else:
            Print(c)
Step 5: Stop.
```

SOURCE CODE

```
print("Enter the three numbers.")
a=int(input("\nEnter the first number: "))
b=int(input("\nEnter the second number: "))
c=int(input("\nEnter the third number: "))
if a>b and a>c:
    print(a," is biggest number.")
elif b>a and b>c:
    print(b,"is biggest number.")
else:
    biggest=c
print(c,"is the biggest number.")
```

OUTPUT

```
Enter the three numbers.

Enter the first number: 45

Enter the second number: 75

Enter the third number: 24
75 is biggest number.
```

RESULT

Program to find biggest of three numbers has been executed successfully and the output is verified.

PageNo:22 **EXPERIMENT NO:11**
Date:05/12/2022 **PROGRAM NAME :FILE EXTENSION**

AIM

Accept a filename from user and print extension of that.

ALGORITHM

```
Step 1: Start.
Step 2: Input a file name.
Step 3: Store the file name extension using split().
Step 4: Print extension.
Step 5: Stop.
```

SOURCE CODE

```
filename=input("Enter the file name with extension: ")
extension=filename.split(".")
print("Extension of given file is ",extension[-1])
```

OUTPUT

```
Enter the file name with extension: programming.py
Extension of given file is  py
```

RESULT

The program to get file extension has been executed successfully, and the output is obtained.

PageNo:23

Date:05/12/ 2022

EXPERIMENT NO:12

PROGRAM NAME:DISPLAY COLORS FROM LIST

AIM

Create a list of colors from comma-separated color names entered by user.Display first and last colors.

ALGORITHM

```
Step 1: Start.
Step 2: Input a list containing name of colors.
Step 3: Store first and last element of list value and print.
Step 4: Stop.
```

SOURCE CODE

```
color_list=["white","green","blue","red","black"]
print("First and Last colors from the list are",color_list[0],"and",
     color_list[-1])
```

OUTPUT

```
First and Last colors from the list are white and black
```

RESULT

Program to display first and last colors has been executed successfully and output is verified.

PageNo:24

Date:05/12/ 2022

EXPERIMENT NO:13
PROGRAM NAME: COMPUTE EXPRESSION

AIM

Accept an integer n and compute n+nn+nnn.

ALGORITHM

```
Step 1: Start
Step 2: Input an integer.
Step 3: Compute the expression n+nn+nnn
Step 4: Print the computed value.
```

SOURCE CODE

```
i=int(input("Enter the number to compute: "))
num=(i+(i*10)+i)+((i*100)+(i*10)+i)
print("The computed value is",num)
```

OUTPUT

```
Enter the number to compute: 5
The computed value is 615
```

RESULT

Program to compute a function has been executed successfully and output is verified.

PageNo:25
Date:05/12/2022

EXPERIMENT NO:14
PROGRAM NAME: DISPLAY DIFFERENCE OF TWO LISTS

AIM

Print out all colors from color_list1 not contained in color_list2.

ALGORITHM

```
Step 1: Start.
Step 2: Input two list with elements as set().
Step 3: Using difference() .
Print list of elements.
Step 4: Stop.
```

SOURCE CODE

```
color_list1=set(["White","Black","Pink","Red"])
color_list2=set(["Black","Green","Blue","Red"])
print("colors from color_list1 not contained in color_list2 is",
      color_list1.difference(color_list2))
```

OUTPUT

```
colors from color_list1 not contained in color_list2 is {'White', 'Pink'}
```

RESULT

Program to display difference of lists has been executed successfully and output is verified.

PageNo:26

Date:05/12/2022

EXPERIMENT NO:15

PROGRAM NAME: SWAP CHARACTER

AIM

Create a single string separated with space from two strings by swapping the character at position 1.

ALGORITHM

Step 1: Start.
Step 2: Define function to swap.
Step 3: Store each string to new variable.
Step 4: Print resulted string.
Step 5: Input two string.
Step 6: Call the function.
Step 7: Stop.

SOURCE CODE

```
def charswap(a, b):
 new_a = b[:1] + a[1:]
 new_b = a[:1] + b[1:]
 return new_a + ' ' + new_b
a=input("Enter string 1 ")
b=input("Enter string 2 ")
print("Swapped strings\n")
print(charswap(a,b))
```

OUTPUT

```
Enter string 1 Hello
Enter string 2 Python
Swapped strings

Pello Hython
```

RESULT

The program to swap first string character has been executed successfully and the output is verified.

Date:08/12/2022

EXPERIMENT NO:16

PROGRAM NAME: SORT DICTIONARY

AIM

Sort dictionary in ascending and descending order.

ALGORITHM

```
Step 1: Input a dictionary.
Step 2: Convert a given dictionary to list, l.
Step 3: To sort in ascending order.
Call l.sort()
For descending order.
Call l.sort(reverse=True)
Step 4: dict=dict(1).
Step 5: Stop.
```

SOURCE CODE

```
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print("Original dictionary : ",d)
list1=list(d.items())
 #convert the given dict. into list
list1.sort()             #sort the list
print("Ascending order is ",list1)
list1=list(d.items())
list1.sort(reverse=True)    #sort in reverse order
print("Descending order is ",list1)
dict=dict(list1) # convert the list in dictionary
print("Dictionary ",dict)
```

OUTPUT

```
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
Ascending order is  [(0, 0), (1, 2), (2, 1), (3, 4), (4, 3)]
Descending order is  [(4, 3), (3, 4), (2, 1), (1, 2), (0, 0)]
Dictionary  {4: 3, 3: 4, 2: 1, 1: 2, 0: 0}
```

RESULT

The Program to sort dictionary has been executed successfully and output is verified.

Date:08/12/2022

EXPERIMENT NO:17
PROGRAM NAME: MERGE DICTIONARY

AIM

Merge two dictionaries.

ALGORITHM

```
Step 1: Start.
Step 2: Input 2 dictionaries .
Step 3: Merge two dictionaries using -Call update() .
Step 5: Print the dictionary after merging.
Step 6: Stop.
```

SOURCE CODE

```
dict1={"Name":"Ruby","Age":"23","Marks":"95"}
dict2={"Dob":"25/10/99","Reg.no":"1234567"}
dict1.update(dict2)
print(dict1)
```

OUTPUT

```
{'Name': 'Ruby', 'Age': '23', 'Marks': '95', 'Dob': '25/10/99', 'Reg.no':
'1234567'}
```

RESULT

Program to merge two dictionaries has been executed successfully and output is verified.

Date:08/12/2022

EXPERIMENT NO:18

PROGRAM NAME: FIND GCD OF TWO NUMBERS

AIM

Find gcd of 2 numbers.

ALGORITHM

```
Step 1: Start.
Step 2: Define function.
Step 3: Check if x>y set
s=y
else set
s=x
Step 4: Check until 1,s+1
if ((x%i==0) and (y%i==0))
Print gcd
Step 5: Call function.
Step 6: Stop.
```

SOURCE CODE

```
def compute_gcd(x,y):
 if x>y:
  s=y

 else:
  s=x

 for i in range(1,s+1):
    if ((x%i==0) and (y%i==0)) :
       gcd=i
 print("GCD of",a,"and",b, "is",gcd)
a=int(input("Enter the a value to find GCD: "))
b=int(input("Enter the another value to find GCD: "))
compute_gcd(a,b)
```

OUTPUT

```
Enter the a value to find GCD: 24
Enter the another value to find GCD: 18
GCD of 24 and 18 is 6
```

RESULT

Program to find GCD has been executed successfully and the output is obtained.

PageNo:33
Date:08/12/2022

EXPERIMENT NO:19
PROGRAM NAME:REMOVE EVEN NUMBERS FROM LIST

AIM

From a list of integers , create a list removing even numbers.

ALGORITHM

```
Step 1: Start.
Step 2: Input a limit to the list,n.
Step 3: Append element to the list until 0,n.
Step 4: Check, for i in list1.
i%2!=0 then,
Append to list even.
Step 5: Print even.
Step 6: Stop.
```

SOURCE CODE

```
list1=[]
even=[]
n=int(input("Enter the limit of list: "))
print("Enter the integers into the list ")
for i in range(0,n):
 print("Enter the element no:{}".format(i+1))
 k=int(input())
 list1.append(k)
print("List of integers are ",list1)
for i in list1:
 if i%2!=0:
  even.append(i)
print("List of integers removing even numbers are ",even)
```

OUTPUT

```
Enter the limit of list: 7
Enter the integers into the list
Enter the element no:1
5
Enter the element no:2
2
Enter the element no:3
17
Enter the element no:4
23
Enter the element no:5
16
Enter the element no:6
19
Enter the element no:7
22
List of integers are  [5, 2, 17, 23, 16, 19, 22]
List of integers removing even numbers are  [5, 17, 23, 19]
```

RESULT

Program to display list of integers removing even numbers has been executed successfully and the output is obtained.