**KGiSL INSTITUTE OF TECHNOLOGY: COIMBATORE**

(Approved by AICTE & Affiliated to Anna University, Chennai)

**GE8161**

**Problem Solving and**

**Python Programming Laboratory**

LAB MANUAL

Department of Computer Science and Engineering

**KGiSL INSTITUTE OF TECHNOLOGY**

Saravanampatti, Coimbatore-35

**GE8161 PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY**

**L T P C**

**0 0 4 2**

**OBJECTIVES:**

* To write, test, and debug simple Python programs.
* To implement Python programs with conditionals and loops.
* Use functions for structuring Python programs.
* Represent compound data using Python lists, tuples, dictionaries.
* Read and write data from/to files in Python.

**LIST OF PROGRAMS:**

1. Compute the GCD of two numbers.
2. Find the square root of a number (Newton‘s method)
3. Exponentiation (power of a number)
4. Find the maximum of a list of numbers
5. Linear search and Binary search
6. Selection sort, Insertion sort
7. Merge sort
8. First n prime numbers
9. Multiply matrices
10. Programs that take command line arguments (word count)
11. Find the most frequent words in a text read from a file
12. Simulate elliptical orbits in Pygame
13. Simulate bouncing ball using Pygame

**OUTCOMES:**

**Upon completion of the course, students will be able to:**

* Write, test, and debug simple Python programs.
* Implement Python programs with conditionals and loops.
* Develop Python programs step-wise by defining functions and calling them.
* Use Python lists, tuples, dictionaries for representing compound data.
* Read and write data from/to files in Python.

**Ex No: 1** COMPUTE THE GCD OF TWO NUMBERS

**AIM**

Write a Python program to compute the greatest common divisor (***gcd***) of two positive integers.

Sample Input1: 17, 17

Sample Output1: 17

Sample Input2: 0, 17

Sample Output2: 17

Sample Input3: 12, 17

Sample Output3: 1

Sample Input3: 44, 33

Sample Output3: 11

**ALGORITHM**

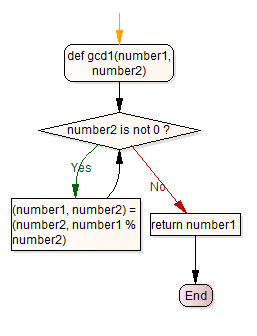
Function: **GCD**

**Parameters**: number1, number2

Step 1: while number2 is not 0,   
 Step 1a: number1, number2 = number2, number1 % number2

Step 2: return number1 as GCD

**FLOWCHART**



**PRE-LAB QUESTIONS**

1. Show your manual workings for the ***gcd*** of 924 and 2562.
2. What is the practical use of calculating the ***gcd*** of two numbers?
3. If you have written the recursive solution of the Euclid algorithm, then write the iterative solution, and vice versa.
4. How does the algorithm for the gcd of two numbers work? What is the name for this algorithm? Why is it so special?
5. Why is the global keyword used in the function get\_twonumbers?
6. Is the function get\_twonumbers() fruitful or not fruitful?
7. Define a function is\_valid\_year with parameter year. The function must return True if the value of year is between 1900 and 3000 (*inclusive*). Otherwise, it must return False.
8. <http://cloudcoder.kgkite.ac.in/cloudcoder/#exercise?c=80,p=6941>

**SOURCE CODE**

def gcd(number1, number2):

while True:

if number2 != 0:

(number1, number2) = (number2, number1 % number2)

else:

break

return number1

def get\_twonumbers():

global a, b

a = input('Enter number: ')

b = input('Enter number: ')

a, b = int (a), int (b)

# Program starts here

a = None # initializing variables

b = None

get\_twonumbers() # to get values from user

gcdval = gcd(a, b)

print ('The gcd is ', gcdval)

**OUTPUT**

Enter number: 12

Enter number: 24

The gcd is 12

**RESULT**

Thus the python program to compute the greatest common divisor (***gcd***) of two positive integers has been written, executed and verified successfully.

**POST-LAB QUESTIONS**

1. What if a or/and b are negative integers? How will you modify the program to handle this? Clue: Use the abs()function.
2. How will find the gcd of three integers?
3. How do you calculate the number of days in a month?

**REFERENCE**

* How to find the gcd of two numbers using prime factorization? <http://j.mp/gcdPrime>
* Why 1 is the common factor? <http://j.mp/gcdOne>
* Recursion: <http://j.mp/gcdDemo>

**Ex No: 2** THE SQUARE ROOT

**(Newton’s Method)**

**AIM**

Write a Python program to Implement a python program that determines the square root of a number using the Newton's method..

Sample Input1: 25

Sample Output1: 5

Sample Input2: 9

Sample Output2: 3

Sample Input3: 20

Sample Output3: 4.47213595

**NEWTON’S METHOD**

The most commonly used approximation algorithm is usually attributed to Isaac Newton. It is typically called Newton’s method, but is sometimes referred to as the Newton-Raphson method.15 It can be used to find the real roots of many functions, but we shall look at it only in the context of finding the real roots of a polynomial with one variable. The generalization to polynomials with multiple variables is straightforward both mathematically and algorithmically.

A polynomial with one variable (by convention, we will write the variable as x) is either zero or the sum of a finite number of nonzero terms, e.g., 3x2 + 2x + 3.

Each term, e.g., 3x2, consists of a constant (the coefficient of the term, 3 in this case) multiplied by the variable (x in this case) raised to a nonnegative integer exponent (2 in this case). The exponent on a variable in a term is called the degree of that term. The degree of a polynomial is the largest degree of any single term. Some examples are, 3 (degree 0), 2.5x + 12 (degree 1), and 3x2 (degree 2). In contrast, 2/x and x0.5 are not polynomials. If p is a polynomial and r a real number, we will write p(r) to stand for the value of the polynomial when x = r. A root of the polynomial p is a solution to the equation p = 0, i.e., an r such that p(r) = 0. So, for example, the problem of finding an approximation to the square root of 24 can be formulated as finding an x such that x2 – 24 ≈ 0. Newton proved a theorem that implies that if a value, call it guess, is an approximation to a root of a polynomial, then guess – p(guess)/p’(guess), where p’ is the first derivative of p, is a better approximation.

For any constant k and any coefficient c, the first derivative of cx2 + k is 2cx. For example, the first derivative of x2 – k is 2x. Therefore, we know that we can improve on the current guess, call it y, by choosing as our next guess y - (y2 - k)/2y. This is called **successive approximation.**

**ALGORITHM**

Function: **square\_root**

**Parameters**: number

Step 1: Initialize  
 epsilon = 0.01

k = number

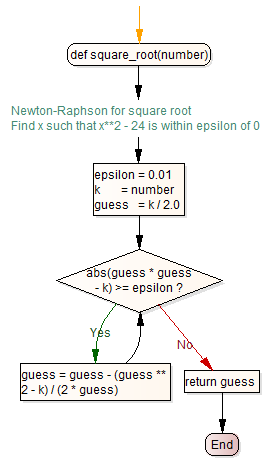
guess = k / 2.0

Step 2: while (abs(guess \* guess- k) >= epsilon) :

guess = guess - (guess \*\*2 - k) / (2 \* guess)

Step 3: return guess

**FLOWCHART**



**PRE-LAB QUESTIONS**

1. What is the square root of 4? of 9? of 20? (Use your calculator if you have to)
2. Manually work out the square root of 20 using the Newton method and show your workings.

**SOURCE CODE**

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

def square\_root(number):

epsilon = 0.01

k = number

guess = k / 2.0

while abs(guess \* guess - k) >= epsilon:

guess = guess - (guess \*\* 2 - k) / (2 \* guess)

return guess

# Program starts here

user\_number = int(input('Input number: '))

sqroot\_number = square\_root(user\_number)

print ('The square root of ', user\_number, 'is about ', sqroot\_number)

**OUTPUT**

Input number: 2

The square root of 2 is about 1.4166666666666667

**RESULT**

Thus the python program to compute the square root of a number using Newton’s method has been written, executed and verified successfully.

**POST-LAB QUESTIONS**

1. Why is the guess initialized to guess = k / 2.0 and not simply guess = k / 2?
2. There is an in-built function for square root calculation? What is it? Compare it with the value that you have generated through your own program.

**Ex No: 3** EXPONENTIATION

**(POWER OF A NUMBER)**

**AIM**

Write a Python program to compute the power of a given number.

Sample Input1: 2,3

Sample Output1: 8

Sample Input2: 0, 3

Sample Output2: 0

Sample Input3: 3,0

Sample Output3: 1

Sample Input4: 2, -3

Sample Output4: 0.125

Sample Input4: 2, "ab"

Sample Output4: Invalid Input

**ALGORITHM**

Function: **EXPO**

**Parameters**: base, expo

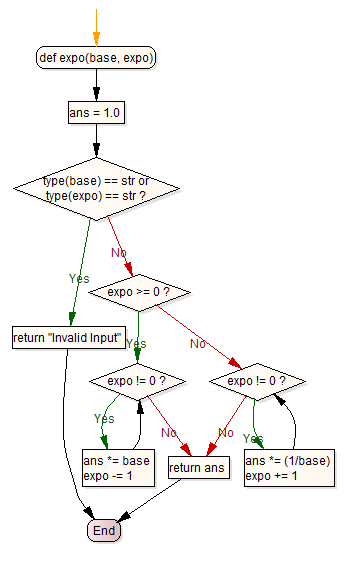
Step 1: Initialize  
 ans = 1.0

Step 2: if the input parameters are not valid,   
 return “invalid input”  
Step 3: if expo is positive number (>=0),  
 multiply base expo times and store it in ans

else,  
 multiply 1/base expo times and store it in ans

Step 4: return the ans

**FLOWCHART**



**SOURCE CODE**

def expo(base, expo):

ans = 1.0

if type(base) == str or type(expo) == str:

return "Invalid Input"

elif expo >= 0:

for i in range(1, expo + 1):

ans = base\*ans

else:

for i in range(expo, 0):

ans = 1/base\*ans

return ans

**OUTPUT**

print(expo(4,3))

64.0

**RESULT**

Thus the python program to compute the number to the power of expo (***exponentiation***) has been written, executed and verified successfully.

**Ex No: 4** FIND THE MAXIMUM OF A LIST OF NUMBERS

**AIM**

Implement a python program that finds the maximum in a list of numbers. The program must call a python function get\_maxnumber that takes as argument a list of numbers (maximum of 10 integers). It must then return the maximum number in that list.

Sample Input1: [5]

Sample Output1: 5

Sample Input2: [1,2]

Sample Output2: 2

Sample Input3: [1,2,3,4]

Sample Output3: 4

**ALGORITHM**

Function: **get\_maxnumber**

**Parameters**: numbers (list)

Step 1: initialize the maximum value

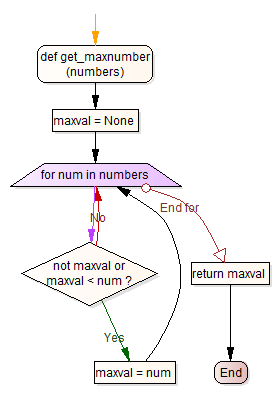
Step 2: Loop through the entries in the list

Step 2a: If maxval is not set, update with number

Step 2b: If maxval is less than number, update with number

Step 3: Return the max value

**FLOWCHART**



**No**

**PRE-LAB QUESTIONS**

1. If the list is a sorted list in ascending order, for e.g. [1, 2, 3, 4, 5, 6], what is the maximum value? What is the python code that will access the maximum value in the list?
2. What is the single line python code that will return the maximum value any list that is sorted in the ascending order?
3. If the list is a sorted list in descending order, for e.g. [5, 4, 3, 2, -1], what is the maximum value? What is the python code that will access the maximum value in the list?
4. What is the single line python code that will return the maximum value in any list that is sorted in the descending order?
5. Assume that the list is already sorted, but it is not known whether it is in ascending order or descending order. In that case, what is the single line python code that will return the maximum value in the list?
6. What are some of the methods that are available to the list python data type?
7. What is the output of the following python code?
8. for x in [1, 3, 5, 7]:

print (x)

1. What is the output of the following python code?
2. for elem in [ 1, 2, 3, 'abc', 99]:

print (elem\*2)

1. What is the code for printing the elements of a list in the reverse order, without actually modifying the list in any way?
2. What is the python code for outputting all the odd integers from 3 to 44?
3. A list contains n elements (where n is a positive integer and 0 > n > 10. Write the necessary python code to produce a list that contains only the last n-1 elements. Is there a version of the code that does not use any list methods whatsoever to achieve the same result?
4. Complete the Hackerrank problem - <https://www.hackerrank.com/challenges/python-lists>

**SOURCE CODE**

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

def get\_maxnumber(numbers):

"""returns the maximum number in the list

....The sequence of steps in the algorithm is:

....#1 - initialize the maximum value

....#2 - Loop through the entries in the list

........#3 - If maxval is not set, update with number

........#3 - If maxval is less than number, update

....#4 - Return the max value

...."""

maxval = None # 1

for num in numbers: # 2

if not maxval or maxval < num: # 3

maxval = num

return maxval # 4

def get\_list\_of\_numbers():

"""returns a list containing elements entered by user

....The sequence of steps in the algorithm is:

....#1 - initialize the list

....#2 - Start loop for maximum of 10 entries

........#3 - Get input from user. If not an number, break

........#4 - Append the number to the list

....#5 - Return the list containing numbers

...."""

ilist = [] # 1

for x in range(0, 10): # 2

try: # 3

userval = input('Enter number ' + str(x) + ': ')

ilist.append(int(userval)) # 4

except ValueError:

break # if user enters a non-integer

return ilist # 5

# Program starts here

ulist = get\_list\_of\_numbers()

print ('List of numbers: ', ulist)

maxnum = get\_maxnumber(userList)

print ('The largest number is ', maxnum)

**OUTPUT**

Enter number 0: 3

Enter number 1: 5

Enter number 2: 6

Enter number 3: 4

Enter number 4: 2

List of numbers: [3, 5, 6, 4, 2]

The largest number is 6

**RESULT**

Thus the python program to find the maximum number from the given list has been written, executed and verified successfully.

**POST-LAB QUESTIONS**

1. Both the functions get\_maxnumber and get\_list\_of\_numbers are not **fruitful** functions. True or False?
2. How will you find the minimum of the values in a list of numbers? What change will you make to the code?
3. If the list is empty, what will be the value returned from the function?
4. If the list happens to be empty, then get\_maxnumber function must return the string "N.A.".. What changes will you make to the code?
5. Sample Input: []

Sample Output: "N.A."

1. For Point 2, refactor the code that you have written using a ternary operator.
2. Write the recursive version for finding the maximum value in a list.
3. How can you create a static variable inside a python function? Give an example.

**ADDITIONAL EXERCISES**

**Sequence structure**

1. Write a program to display “Hello World”
2. Write a program to greet yourself ( eg: “Hello Ashok”)
3. Get and display your name, date of birth and phone number.
4. Print ‘F’ using hash (#), where the F has a height of six characters and width of five and four characters.  
     
   ######

#

#

#####

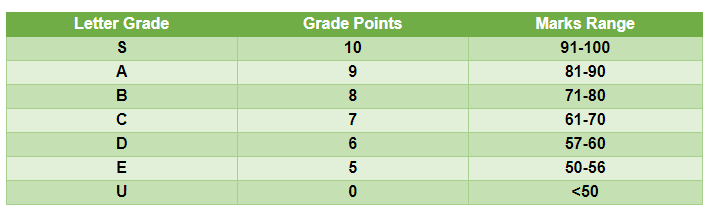
#

#

#

1. Get and print three characters in reverse order  
   ‘ABC’ as ‘CBA’
2. Write a program to double a given number
3. Write a program to add two numbers
4. Write a program to divide two numbers
5. Write a program to find the sum of three numbers
6. Write a program to find the average of three numbers
7. Write a program to find the area of the rectangle
8. Write a program to find the area of the square
9. Write a program to find the area of the triangle
10. Write a program to find y, given x, using the following relation.  
     
11. Write a program to convert the temperature from Celsius to Fahrenheit   
     
12. Write a program to do the following:
    1. Assign any two numbers to x and y
    2. Assign the sum of x and y to z
    3. Output the values of x, y and z
13. Assume that the distance between KITE and Gandhipuram is 12 km. How long will you take to reach Gandhipuram, if
    1. you walk @ 3 km/hour
    2. you travel by cycle @ 10 km/hour
    3. you travel by bike @ 45 km/hour
14. Assume that the distance between KITE and Gandhipuram is 12 km. You walk for 10 minutes @ 10 km/hour. Then you travel by cycle for 10 minutes @ 10 km/hour. Finally, you travel by bike @ 45 km/hour till you reach your destination. How long will you take to reach Gandhipuram.
15. Convert the time in hours to minutes. (eg: 1.25 hour = 75 minutes)
16. Express your age in terms of
    1. months
    2. days
    3. hours   
       (ignore leap years)
17. Solve the Quadratic equation  
    

**Selection structure**

1. Find the maximum of two numbers
2. Write the program to print PASS/FAIL for the given score  
   (eg: score = 54 print “PASS”)  
   rule: score >= 50 “PASS”
3. Find the grade for the given score, based on Anna university Credit system  
   
4. If you purchase for more than Rs 1000, you will get the discount of 10%. If it exceeds Rs 5000, you will get the additional discount of 5%. Write the program to calculate the total after discount for the given purchased amount.

**Modulo operator**

1. Write a function check(N, M) to check whether N is the multiple of M

eg: N = 122 M=3 returns False

1. Write a function add\_rem\_235(N) to add all the reminders when N is divided by 2, 3 and 5 respectively.

Write a function `add\_rem\_235(N)` to add all the reminders

when N is divided by 2, 3 and 5 respectively.

eg:

N = 122

N % 2 = 0

N % 3 = 2

N % 5 = 2

The function returns 0 + 2 + 2 = 4

1. Write a function isEven(N) to find whether N is even or not.

eg: N = 9 returns True

1. Write a function getEven(N) to print even numbers upto N.

eg: N = 10 returns [2, 4, 6, 8, 10]

1. Write a function sum\_digits(N) to find the sum of all digits of N.

eg: N = 12345 sum = 1 + 2 + 3 + 4 + 5 = 9

1. Write a function sum\_odd\_digits(N) to find the sum of all the digits in the odd positions of N.

eg: N = 12345 sum\_odd\_digits = 1 + 3 + 5 = 9

1. Write a function ceil10(N) to round the given N to the nearest **upper** multiple of 10s.

eg: N = 122 ceil10 = 130

1. Write a function floor10(N) to round the given N to the nearest **lower** multiple of 10s.

eg: N = 122 floor10 = 120

1. Write a function round10(N) to round the given N to the **nearest** multiple of 10s.

eg: N = 122 round10 = 120

1. Write a function dec2bin(N) to convert the decimal number N to its binary representation (in string).

eg: N = 12 binary\_equivalent = 'b1100'

1. Write a function to check whether given year is leap or not.

* A year is a leap year, if it is divisible by 4 and not by 100
* A year is a leap year, if it is divisible by 400

Example:

isLeap(2000) --> True

isLeap(2100) --> False

isLeap(2016) --> True

1. Write a function to find the number of days for the given month and the year.

Note: You need to find out whether given year is leap or not, before finding the number of days.

Example:

days(2, 2012) --> 29 # as 2012 is the leap year

days(4, 2010) --> 30

days(10, 2017) --> 31

1. Find the GCD of the sums of digits of two numbers.

Example:

input: 234, 456

sum1 = 2 + 3 + 4 = 9 sum2 = 4 + 5 + 6 = 15

gcd(sum1,sum2) = 3

1. There are 'x' male students and 'y' female students studying in a class. We need to form students teams of equal size. Teams shall not have the mix of boys and girls. The team size shall not exceed 5. Write a function to find the minimum number of teams that can be formed.

ADDITIONAL CLARIFICATIONS:

* When forming teams, no student must be left behind, meaning every student must be assigned to a team.
* The team size of both females and males chosen must not be be the same.

Example: In a classroom, there are '10' boys and '20' girls.

Condition:

1. Teams should be of equal size.

2. The team size shall not exceed 5.

For a team of '5' members, we will have 6 teams. For a team of '2' members, we will have 15 teams. For a team of '10' members, we will have 3 teams.

As team size cannot exceed 5, you need to return '6' as the minimum teams that can be formed with equal size.

**Functions**

# Python Program to find the LCM of Two Numbers

# Python Program to find the GCD of Two Numbers

# Python Program to check Whether a Given Year is a Leap Year

# Python Program to Read Height in Centimeters and then Convert the Height to Feet and Inches

# Python Program to find the Smallest Divisor of an Integer

# Python Program to Take in the Marks of 5 Subjects and Display the Grade

# Python Program to find Those Numbers which are Divisible by 7 and Multiple of 5 in a Given Range of Numbers

# Python Program to print all the Prime Numbers within a Given Range

# Python Program to print Numbers in a Range (1,upper) Without Using any Loops

# Python Program to find the Area of a Triangle Given All Three Sides

# Python Program to check if a Number is a Prime Number

# Python Program to check if a Number is an Armstrong Number

# Python Program to check if a Number is a Perfect Number

# Python Program to find the Fibonacci Series without Using Recursion

# Python Program to find the factorial of a number without recursion

# Python Program to find the Power of a Number Using Recursion

# Python Program to find the Fibonacci Series Using Recursion

# Python Program to find the LCM of Two Numbers Using Recursion

# Python Program to find the GCD of Two Numbers Using Recursion

# Python Program to find the Product of two Numbers Using Recursion

# Python Program to check if a Number is odd or even

# Python program to check if a number is positive, negative or zero

# Python Program to Read Two Numbers and Print Their Quotient and Remainder

# Python Program to find the Gravitational Force Acting between Two Objects

# Python Program to determine whether a given number is even or odd recursively

**List**

1. Given an array of ints, return True if 6 appears as either the first or last element in the array. The array will be length 1 or more.
2. Given an array of ints, return True if the array is length 1 or more, and the first element and the last element are equal.
3. Given an array of ints length 3, return the sum of all the elements.
4. Given an array of ints length 3, return an array with the elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.
5. Given an array of ints length 3, return a new array with the elements in reverse order, so {1, 2, 3} becomes {3, 2, 1}.
6. Given an array of ints length 3, figure out which is larger, the first or last element in the array, and set all the other elements to be that value. Return the changed array.
7. Given an array of ints, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, returning 0 if the array is length 0.
8. Given 2 int arrays, a and b, each length 3, return a new array length 2 containing their middle elements.
9. Given an array of ints, return a new array length 2 containing the first and last elements from the original array. The original array will be length 1 or more.
10. Given an int array length 2, return True if it contains a 2 or a 3.
11. Return the number of even ints in the given array.
12. Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array.
13. Return the "centered" average of an array of ints, which we'll say is the mean average of the values, except ignoring the largest and smallest values in the array. If there are multiple copies of the smallest value, ignore just one copy, and likewise for the largest value. Use int division to produce the final average. You may assume that the array is length 3 or more.

Eg: centered\_average([-10, -4, -2, -4, -2, 0]) → -3

1. Return the sum of the numbers in the array, returning 0 for an empty array. Except the number 13 is very unlucky, so it does not count and numbers that come immediately after a 13 also do not count.

Eg: sum13([1, 2, 2, 1, 13]) → 6

1. Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 7 (every 6 will be followed by at least one 7). Return 0 for no numbers.

Eg: sum67([1, 1, 6, 7, 2]) → 4

1. Given an array of ints, return True if the array contains a 2 next to a 2 somewhere.
2. Write a Python program to sum all the items in a list.
3. Write a Python program to multiplies all the items in a list.
4. Write a Python program to get the largest number from a list.
5. Write a Python program to get the smallest number from a list.
6. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.    
   Sample List : ['abc', 'xyz', 'aba', '1221']  
   Expected Result : 2
7. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.

Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]  
Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

1. Write a Python program to remove duplicates from a list.
2. Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.   
   Sample List : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
   Expected Output : ['Green', 'White', 'Black']
3. Write a Python program to print the numbers of a specified list after removing even numbers from it.
4. Write a Python program to get the difference between the two lists.
5. Write a Python program to append a list to the second list.
6. Write a Python program to find the second smallest number in a list.
7. Write a Python program to convert a list of multiple integers into a single integer.   
   Sample list: [11, 33, 50]  
   Expected Output: 113350
8. Write a Python program to split a list every Nth element.  Sample list: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']  
   Expected Output: [['a', 'd', 'g', 'j', 'm'], ['b', 'e', 'h', 'k', 'n'], ['c', 'f', 'i', 'l']]