1. Write a Python Program to find out largest among three numbers.

Algorithm:

Input : num1, num2, num3: To store the three numbers

largest: To store the largest number found.

Output : Printing the largest number.

```
1. Start
2. if  num1  >=  num2
    2.1.  if  num1  >=  num3
          2.1.1.  largest  =  num1
    2.2.  else
          2.2.1.  largest  =  num3
3. else
    3.1.  if  num2  >=  num3
          3.1.1.  largest  =  num2
    3.2.  else
          3.2.1.  largest  =  num3
4. print  larget
5. End
```

Source Code:

```
n1 = int(input("Enter the 1st number: "))

n2 = int(input("Enter the 2nd number: "))

n3 = int(input("Enter the 3rd number: "))

if n1 > n2 and n1 > n3:

   print(f"\n\t{n1} is the largest\n")

elif n2 > n3 and n2 > n3:

   print(f"\n\t{n2} is the largest\n")

else:

   print(f"\n\t{n3} is the largest\n")
```

Output:

```
Enter the 1st number: 6

Enter the 2nd number: 4

Enter the 3rd number: 9


  9 is the largest
```

2. Write a Python Program to check whether a given number is Perfect number or not.

Algorithm:

Input : Take a number from the user.

Output : Print the given number is perfect or not.

```
1. Start
2. For I ← 0 to num do
   2.1. if I%2 = 0 then sum = sum+i
3. if num = sum then print “The number is perfect”
4. else print “The number is not perfect”
5. End
```

Source Code:

```
num = int(input("Enter the number: "))
sum = 0
for i in range(1,num):
    if i % 2 == 0:
        sum += i
if num == sum:
    print(f"\n\t{num} is perfect number\n")
else:
    print(f"\n\t{num} is not perfect number\n")
```

Output:

```
Enter the number: 123
    123 is not perfect number
Enter the number: 6
    6 is perfect number
```

3. Write a Python Program to generate Prime numbers within a given range.

Algorithm:

Input : Take a starting and ending number from the user.

Output : Print the prime numbers from given range.

```
1. Start
2. For I ← start to end do
   2.1. Count ← 0
   2.2. For j ← 2 to I do
        2.2.1. If I%j = 0 then count ← count+1
   2.3. End for
   2.4. If count = 0 then print i
3. End for
4. End
```

Source Code:

```python
start = eval(input("Enter the starting range: "))
end = eval(input("Enter the ending range: "))
print(f"Prim numbers between {start} to {end} is: ",end=" ")
for i in range(start,end):
    count = 0
    for j in range(2,i):
        if i%j == 0:
            count += 1
    if count == 0:
        print(i,end=" ")
```

Output:

```
Enter the starting range: 1
Enter the ending range: 10
Prim numbers between 1 to 10 is:  1 2 3 5 7
```

4. Write a Python Program to print the following patterns:-

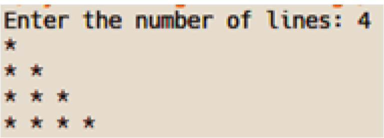
```
*                 *                  *
* *             * * *              * * *
* * *         * * * * *          * * * * *
* * * *     * * * * * * *          * * *
                                     *
```

Source Code:

```
# Pattern 1
num = eval(input("Enter the number of lines: "))
for i in range(num):
    for j in range(i+1):
        print("*",end=" ")
    print()

# Pattern 2
num = eval(input("Enter the number of lines: "))
for i in range(num):
    for k in range(num - i):
        print(" ",end=" ")
    for j in range(2*i + 1):
        print("*",end=" ")
    print()
# Pattern 3
n = int(input("Enter the number of lines: "))


for i in range(n):
    print(" " * (n - i - 1) + "*" * (2 * i + 1))


for i in range(n - 2, -1, -1):
    print(" " * (n - i - 1) + "*" * (2 * i + 1))
```

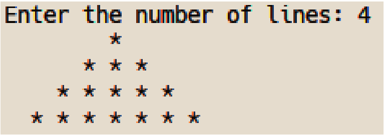
Output:

# Pattern 1



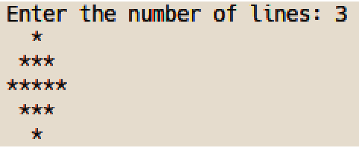
```
Enter the number of lines: 4
*
* *
* * *
* * * *
```

# Pattern 2



```
Enter the number of lines: 4
        *
      * * *
    * * * * *
  * * * * * * *
```

# Pattern 3



```
Enter the number of lines: 3
  *
 ***
*****
 ***
  *
```

5. Write a Python Program to check whether a given number is Palindrome number or not.

Algorithm:

Input : Take a number from the user.

Output : Print that the given number is palindrome or not.

```
1. Start
2. Temp ← num
3. Rev ← 0
4. While temp > 0 do
   4.1. Digit ← temp%10
   4.2. Rev ← rev*10+digit
5. end while
6. If num = rev then print “The number is palindrome”
7. Else print “The number is not palindrome”
8. End
```

Source Code:

```
num = eval(input("Enter the number: "))

temp = num

rev = 0

while temp > 0:

  digit = temp % 10

  rev = rev * 10 + digit

  temp //= 10

if num == rev:

  print("The number is palindrome")

else:

  print("The number is not palindrome")
```

Output:

```
 Enter the number: 121
           The number is palindrome

Enter the number: 124

           The number is not palindrome
```

6. Write a Python Program to check whether a given number is divisible by 11 or not.

Algorithm:

Input : Take the number from the user.

Output : Print the given number is divisible by 11 or not.

```
1. Start
2. If num % 11 = 0 then print that the number is divisible by 11
3. Else print that the number is not divisible by 11
4. End
```

Source Code:

```
num = eval(input("Enter a number: "))
if num % 11 == 0:
   print("The number is divisible by 11")
else:
   print("The number is not divisible by 11")
```

Output:

```
Enter a number: 121
The number is divisible by 11
Enter a number: 10
The number is not divisible by 11
```

7. Write a Python Program to calculate GCD and LCM of two given numbers.

Algorithm:

Input : Take two number form the user.

Output : Print the GCD and LCM of that numbers.

```
1. Start
2. X ← num1
3. Y ← num2
4. While num2 ≠ 0
    4.1. Temp ← num2
    4.2. Num2 ← num1%num2
    4.3. Num1 ← temp
5. end while
6. gcd ← num1
7. lcm ← (num1*num2)/gcd
8. print gcd,lcm
9. End
```

Source Code:

```
a = eval(input("Enter the 1st number: "))
b = eval(input("Enter the 2nd number: "))
x = a
y = b
while b != 0:
   temp = b
   b = a % b
   a = temp
gcd = a
lcm = (x * y) / gcd
print("The GCD of", x, "and", y, "is", gcd)
print("The LCM of", x, "and", y, "is", lcm)
```

Output:

```
Enter a number: 121

The number is divisible by 11

Enter a number: 10

The number is not divisible by 11
```

8. Write a Python Program to check whether a given number is Armstrong number or not.

Algorithm:

Input : Take a number from the user.

Output : Print the given number is Armstrong or not.

```
1. Start
2. temp ← num
3. sum ← 0
4. while temp $\neq$ 0 do
   4.1. digit ← temp%10
   4.2. temp ← temp/10
   4.3. sum ← sum+digit$^3$
5. if sum = num then print “The number is Armstrong number”
6. else print “The number is not Armstong number”
7. End
```

Source Code:

```
num = eval(input("Enter the number: "))
temp = num
sum = 0
while temp != 0:
   digit = temp % 10
   temp = temp // 10
   sum += pow(digit,3)
if sum == num:
   print("\n\tThe number is armstrong number\n")
else:
   print("\n\tThe number is not armstrong number\n")
```

Output:

```
Enter the number: 153

   The number is Armstrong number

Enter the number: 123

   The number is not Armstrong number
```

9. Write a Python Program to take a input name with surname. Display the input in abbreviated form.

Algorithm:

Input : Take a name as string from the user.

Output : Print the abbreviated form for given name.

```
1. Start
2. First we split words from the user given string in to a list
3. abb ← ‘ ’
4. for each I in words list do
    4.1. if the first words first letter is capital ?
        4.1.1. convert the first letter into upper case and append with
               ‘abb’
    4.2. else append the first letter with ‘abb’
    4.3. end if
    4.4. abb ← ‘ . ’
5. end for
6. End
```

Source Code:

```
string = str(input("Enter the string: "))
words = string.split()
abb = ''
for i in words:
    if not i[0].isupper():
        abb += i[0].upper()
    else:
        abb += i[0]
    abb += '.'
print(f"The abbreviation form of your string: {abb}")
```

Output:

```
Enter the string: Anit Halder
The abbreviation form of your string: A.H.
```

10. Write a Python Program to print the longest word from a text of line.

Algorithm:

Input : Take a string from the user.

Output : Print the longest word from the text of line.

1. Start
2. Split the user given string into list of words
3. Making a list which include length of each words
4. Find the maximum word from words lengths list and print that word
5. End

Source Code:

```
string = str(input("Enter the string: "))

s = string.split()

l = [len(i) for i in s]

print(f"\nThe longest word from line of text is: {s[l.index(max(l))]} \n")
```

Output:

```
Enter the string: I am anit halder

The longest word from line of text is: halder
```

11. Write a Python Program to print the longest line from a text of line.

Algorithm:

Input : Take n which include number of string user want and take user given string.

Output : Print the longest line from a text of line.

```
1. Start
2. For I ← 0 to n do
   2.1.  Take user given string and store it into a list
3. End for
4. For I in list of strings then do
   4.1.  Find the length of the string and store it into a list
5. End for
6. Find the largest from the list which include length of each string and
   print it
7. End
```

Source Code:

```python
num = eval(input("Enter the how many number of lines you want to read: "))
string = []
for i in range(num):
  string.append(str(input("Enter the string: ")))
l = [len(i) for i in string]
print(f"\nThe longest word from line of text is: {string[l.index(max(l))]} \n")
```

Output:

```
Enter the how many number of lines you want to read: 3
Enter the string: Myself Anit Halder
Enter the string: It's my last year in college
Enter the string: I am a very bad guy
The longest word from line of text is: It's my last year in college
```

12. Write a Python Program to check whether a given particular word in a text of lines is present or not.

Algorithm:

Input : Take n number of line from the user store it in ‘strings’ which is list and take the word which you want to check.

Output : Print the “word is present” if the given word is present.

```
1. Start
2. Flag ← false
3. For I is present in strings where I is each line of string list
    3.1. If the checking word is present in i
            3.1.1. Flag ← true
    3.2. End if
4. End for
5. If flag = true then print “The word is present in the text of lines”
6. else print “The word is not present in the text of lines”
7. End
```

Source Code:

```
num = eval(input("Enter the how many number of lines you want to read: "))
strings = []
for i in range(num):
  strings.append(str(input("Enter the string: ")))
flag = False
check = str(input("Enter a word you want to find: "))
for i in strings:
  if check in i:
    flag = True
if flag == True:
  print("The word is in the text of lines")
else:
  print("The word is not in the text of lines")
```

Output:

```
Enter the how many number of lines you want to read: 3
Enter the string: Myself anit halder
Enter the string: It's my last year in college
Enter the string: I am a very bad guy
Enter a word you want to find: my
          The word is in the text of lines
```

13. Write a Python program to construct another list L2 which consists the indices of all non-zero elements of list L1.

Algorithm:

Input : Take a list from the user.

Output : Print the list of non-zero element’s index of list L1.

```
1. Start
2. for I in list l1 do
   2.1.   if I ≠ 0 then store the index of that element in another list
3. end for
4. print the list of indices of all non-zero elements of list
5. End
```

Source Code:

```
l = [10, 0, 2, 0, 0, 5, 7]

print("The given list is: ",l)

l = [l.index(i) for i in l if i != 0]

print("The list of indices of all non-zero elements of list: ",l)
```

Output:

```
The given list is:  [10, 0, 2, 0, 0, 5, 7]
The list of indices of all non-zero elements of list:  [0, 2, 5, 6]
```

14. Write a python program to print the maximum length of word from a list of words.

Algorithm:

Input : Take a list of words from the user and store in a list.

Output : Print the word which length is maximum.

```
1. Start
2. Max ← " "
3. Iterate over all elements which present in the list
   3.1. If length of current element is > the length of 'max' then
        replace 'max' content with current element of the list
4. Print the 'max'
5. End
```

Source Code:

```
lis = ['Apple', 'Orange', 'Mango', 'Pineapple']
max = ''
for i in lis:
    if len(i) > len(max):
        max = i
print("The maximum word is: ",max)
```

Output:

```
The given word list is:  ['Apple', 'Orange', 'Mango', 'Pineapple']

The maximum word is:  Pineapple
```

15. Write a python program to print those pair of index of value that is formed a particular number in the given list.

Algorithm:

Input : Take a list with elements with and take the number whose pair you want to find.

Output : Print the pairs of index of user given number.

```
1. Start
2. Create an empty list 'pair-lis' to store pairs of indices
3. For I ← 0 to length of given list
    3.1. For j ← 0 to length of given list
            3.1.1. If lis[i] + lis[j] = num then pair-lis.append((I,j))
    3.2. End for
4. End for
5. Print pair-lis
6. End
```

Source Code:

```
lis = [10, 20, 30, 40, 50, 60, 70]
print("The given list of numbers: ", lis)
num = eval(input("Enter a number: "))
lis = [(lis.index(lis[i]), lis.index(lis[j])) for i in range(len(lis)) for j in range(len(lis))
if lis[i] + lis[j] == num]
print("The pairs are: ", end="")
for i in lis:
  print(i, end=" ")
print()
```

Output:

```
The given list of numbers:  [10, 20, 30, 40, 50, 60, 70]

Enter a number: 60

The pairs are: (0, 4) (1, 3) (2, 2) (3, 1) (4, 0)
```

16. Write a python program to print only even pair elements from a given nested list.

Algorithm:

Input : Take a list and sub-list in that list with elements from the user.

Output : Print the even pairs list.

```
1. Start
2. First take an empty list
3. For each I from the lis
    3.1. Count ← 0
    3.2. For each j from i
            3.2.1. If j%2 = 0 then count ← count + 1
    3.3. End for
    3.4. If count = length of I then newLis.append(i)
4. End for
5. End
```

Source Code:

```
lis = [[2, 6], [3, 8], [10, 14], [1, 10]]
print("The given list of numbers: ", lis)
print("The even pairs are: ", end="")
newLis = []
for i in lis:
  count = 0
  for j in i:
    if j % 2 == 0:
      count += 1
  if count == len(i):
    newLis.append(i)
for i in newLis:
  print(i, end=" ")
print()
```

Output:

```
The given list of numbers:  [[2, 6], [3, 8], [10, 14], [1, 10]]

The even pairs are: [2, 6] [10, 14]
```

17. Write a Python program to rotate the element of the list such that, the element of the first index moves to the second index , the element of the second index moves to the third index ... and at last the element of the last index moves to the first index.

Algorithm:

Input : Take a index from where user want to rotate.

Output : Print the list after rotation.

1. Start
2. Take the index from the user in ‘num’
3. Take a empty list
4. Now take elements of given list from index num-1 to end using slicing operator
5. Take elements of given list from index 0 to num-1 using slicing operator
6. Concatenate that and insert it into the empty list
7. Print that list
8. End

Source Code:

```
lis = [1, 2, 3, 4, 5, 6]
print("The given list of numbers: ", lis)
num = eval(input("Enter the index from where you want to rotate: "))
lis = lis[num - 1:] + lis[:num - 1]
print("The rotated list is: ", lis)
```

Output:

```
The given list of numbers:  [1, 2, 3, 4, 5, 6]

Enter the index from where you want to rotate: 3

The rotated list is:  [3, 4, 5, 6, 1, 2]
```

18. Given two list as L1 and L2. Construct the third list L3 in which the elements should be the value of list L2, followed by the value of list L1 alternatively.

Algorithm:

Input : Take two list with elements from the user and store it into L1 & L2 and take a empty list L3.

Output : Print the elements of L2, followed by the value of list L1 alternatively.

```
1. Start
2. For I ← 0 to length of the list which length is greater
    2.1. If I < length of L1 then append the content of L1[i] with L3
    2.2. If I < length of L2 then append the content of L2[i] with L3
3. Print list L3
4. End
```

Source Code:

```
L1 = [1, 2, 3, 4]
L2 = [10, 20, 30, 40, 50, 60]
L3 = []
print("Input list 1: ", L1)
print("Input list 2: ", L2)
for i in range(max(len(L1), len(L2))):
  if i < len(L1):
    L3.append(L1[i])
  if i < len(L2):
    L3.append(L2[i])
print("Output: ", L3)
```

Output:

```
Input list 1:  [1, 2, 3, 4]
Input list 2:  [10, 20, 30, 40, 50, 60]
Output:  [1, 10, 2, 20, 3, 30, 4, 40, 50, 60]
```

19. Write a Python program to eliminate the duplicate numbers from a list.

Algorithm:

Input : Take a list 'lis' with elements and take a empty list.

Output : Print the updated list.

```
1. Start
2. For each I in list 'lis'
   2.1. If I not present in 'ls' then append I with 'ls'
3. End for
4. Print the given list
5. Print the updated list
6. End
```

Source Code:

```
lis = [1, 2, 3, 3, 2, 1, 4, 3, 5, 6, 5]
ls = []
for i in lis:
  if i not in ls:
    ls.append(i)
print("The given list is: ", lis)
print("The new list is: ", ls)
```

Output:

```
The given list is:  [1, 2, 3, 3, 2, 1, 4, 3, 5, 6, 5]

The new list is:  [1, 2, 3, 4, 5, 6]
```

20. Write a Python Program to store 5 individual marks of n numbers of students into a tuple. Find out the average marks of each student and also print the index of which average comes 1st position.

Algorithm:

Input : Take a tuple with sub tuple which store 5 individual marks of n student in ‘marks’.

Output : Print the index of marks which average is highest.

```
1. Start
2. For I in each element of marks
    2.1. Calculate the sum of all elements from the sub tuple and
         divide it with length of I then append it a blank list L
3. End for
4. Print the index of the highest average marks
5. End
```

Source Code:

```python
marks = ((40, 50, 30, 70), (60, 80, 90, 50), (80, 70, 50, 40), (30, 50, 70,260))
l = []
for i in marks:
  l.append(sum(i) / len(i))
print("The list of marks each student is: ", marks)
print("The index of highest average is: ", l.index(max(tuple(l))))
```

Output:

```
The list of marks each student is:  ((40, 50, 30, 70), (60, 80, 90, 50), (80, 70, 50, 40), (30,   50, 70, 260))
  The index of highest avarage is:  3
```

21. Write a Python Program to check Prime numbers from a tuple and store the index of those prime numbers into another tuple.

Source Code:

```
from sympy import isprime

numbers = (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)

prime_indices = tuple(index for index, num in enumerate(numbers) if

isprime(num))

print("The given lis of numbers: ", numbers)

print("The prime numbers are: ", prime_indices)
```

Output:

```
The given lis of numbers:  (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
The prime numbers are:  (0, 1, 3, 5, 9, 11)
```

22. Write a Python Program to take input of n numbers of students’ names and their total marks. Display the student’s name who comes first.

Source Code:

```
students = []

marks = []

num = int(input("Enter the number of students: "))

for i in range(num):

  name = input("\nEnter the name of the student: ")

  mark = eval(input("Enter the total marks of the student: "))

  students.append(name)

  marks.append(mark)

students = tuple(students)

marks = tuple(marks)


print("\nThe name of the student who comes first:

",students[marks.index(max(marks))])
```

Output:

```
Enter the number of students: 3

Enter the name of the student: anit
Enter the total marks of the student: 300

Enter the name of the student: ajoy
Enter the total marks of the student: 285

Enter the name of the student: arpan
Enter the total marks of the student: 295
The name of the student who come first:  anit
```

23. Given a dictionary d1 = {1:’A’, 2:’B’, 3:’C’} . Write a Python Program to create another dictionary d2 such that the data values of key and value of d1 just reverse of d2.

Source Code:

```
dic = {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

print("The given dictionary is: ", dic)

new_dic = {v: k for k, v in dic.items()}

print("The new dictionary is: ", new_dic)
```

Output:

```
The given dictionary is:  {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

The new dictionary is:  {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
```

24. Write a Python Program that defines a dictionary to store names of 5 cricketers along with their total runs to do the following:

a) Sort the dictionary in ascending order of their runs.

b) Display the name of cricketer with highest runs.

c) Display only the name of cricketers.

d) remove the cricketer having the lowest runs.

Source Code:

```
# Define the dictionary with cricketers and their total runs
cricketers_data = {
    'Player1': 7500,
    'Player2': 6000,
    'Player3': 8200,
    'Player4': 5500,
    'Player5': 6800
}

# a) Sort the dictionary in ascending order of their runs
sorted_cricketers = dict(sorted(cricketers_data.items(), key=lambda x: x[1]))
```

```
# Display the sorted dictionary
print("a) Cricketers sorted by runs:")
for player, runs in sorted_cricketers.items():
    print(f"{player}: {runs} runs")

# b) Display the name of the cricketer with the highest runs
max_runs_player = max(cricketers_data, key=cricketers_data.get)
print("\nb) Cricketer with the highest runs:")
print(f"{max_runs_player}: {cricketers_data[max_runs_player]} runs")

# c) Display only the names of cricketers
print("\nc) Names of Cricketers:")
for player in cricketers_data.keys():
    print(player)

# d) Remove the cricketer having the lowest runs
min_runs_player = min(cricketers_data, key=cricketers_data.get)
del cricketers_data[min_runs_player]

# Display the updated dictionary after removing the cricketer
print("\nd) Dictionary after removing the cricketer with the lowest runs:")
for player, runs in cricketers_data.items():
    print(f"{player}: {runs} runs")
```

Output:

```
a) Cricketers sorted by runs:
Player4: 5500 runs
Player2: 6000 runs
Player5: 6800 runs
Player1: 7500 runs
Player3: 8200 runs
b) Cricketer with the highest runs:
Player3: 8200 runs
c) Names of Cricketers:
Player1
Player2
Player3
Player4
Player5
d) Dictionary after removing the cricketer with the lowest runs:
Player1: 7500 runs
Player2: 6000 runs
Player3: 8200 runs
Player5: 6800 runs
```

25. Write a Python Program that accepts a hyphen-separated sequence of words as input and print the words in also hyphen-separated after sorting them alphabetically.

Source Code:

```
# Accept hyphen-separated sequence of words as input
input_sequence = input("Enter a hyphen-separated sequence of words: ")

# Split the input string into a list of words
words = input_sequence.split('-')

# Sort the list of words alphabetically
sorted_words = sorted(words)

# Join the sorted words into a hyphen-separated string
sorted_result = '-'.join(sorted_words)

# Print the sorted output
print("Sorted Output:", sorted_result)
```

Output:

```
Enter a hyphen-separated sequence of words: Banana-Apple-Orange
Sorted Output: Apple-Banana-Orange
```

26. Write a Python Program to count the lines, words, characters from a text file.

Source Code:

```
# Input: File path
file_path = input("Enter the path of the text file: ")

# Open the file for reading
file = open(file_path, 'r')

# Read the entire content of the file
content = file.read()

# Count lines
lines = content.count('\n') + 1

# Count words
words = len(content.split())

# Count characters
characters = len(content)

# Close the file
file.close()

# Output: Print the results
print(f"\nNumber of Lines: {lines}")
print(f"Number of Words: {words}")
print(f"Number of Characters: {characters}")
```

Output:

# text.txt

```
good morning madam

how are you

i am fine
```

# output

```
Enter the path of the text file: text.txt

Number of Lines: 3

Number of Words: 9

Number of Characters: 40
```

27. Write a Python Program to display the longest line from a text file.

Source Code:

```
# Input: File path
file_path = input("Enter the path of the text file: ")

# Open the file for reading
with open(file_path, 'r') as file:
    # Read all lines from the file
    lines = file.readlines()

    # Find the longest line
    longest_line = max(lines, key=len)

    # Output: Print the longest line
    print("\nThe Longest Line: ",end='"')

    # Strip removes leading and trailing whitespaces
    print(f'{longest_line.strip()}"')
```

Output:

# text.txt

```
good morning madam
how are you
i am fine
```

# output

```
Enter the path of the text file: text.txt
The Longest Line: "good morning madam"
```

28. Write a Python Program that reads numbers from keyboard one by one. All even numbers get stored inside the file called even and all odd number get stored inside the file called even and all odd numbers get stored inside the file called odd and others get stored inside the file others.

Source Code:

```
# Open files for writing using 'with' statement
with open('even.txt', 'w') as even_file, open('odd.txt', 'w') as odd_file,
open('others.txt', 'w') as others_file:
    while True:
        # Input: Read a number from the keyboard
        num_str = input("Enter a number (or 'exit' to stop): ")

        # Check if the user wants to exit
        if num_str.lower() == 'exit':
            break

        # Check if the input can be converted to an integer
        if num_str.isdigit() or (num_str[0] == '-' and num_str[1:].isdigit()):
            # Convert the input to an integer
            num = int(num_str)

            # Categorize the number and store in respective files
            if num % 2 == 0:
                even_file.write(f"{num}\n")
            elif num % 2 != 0:
                odd_file.write(f"{num}\n")
            else:
                others_file.write(f"{num}\n")
        else:
            print("Error: Please enter a valid number.")
```

Output:

```
Enter a number (or 'exit' to stop): 1      Enter a number (or 'exit' to stop): 7
Enter a number (or 'exit' to stop): 2      Enter a number (or 'exit' to stop): 8
Enter a number (or 'exit' to stop): 3      Enter a number (or 'exit' to stop): 9
Enter a number (or 'exit' to stop): 4      Enter a number (or 'exit' to stop): 10
Enter a number (or 'exit' to stop): 5      Enter a number (or 'exit' to stop): exit
Enter a number (or 'exit' to stop): 6
```

```
# odd.txt          # even.txt          # others.txt
   1                   2                   null
   3                   4
   5                   6
   7                   8
   9                   10
```

29. Write a user define function in python to count and display the number of lines starting with alphabet ‘A’ & ‘S’ present in a text file “text.txt”.

Source Code:

```
L1 = [1, 2, 3, 4]
L2 = [10, 20, 30, 40, 50, 60]
L3 = []
print("Input list 1: ", L1)
print("Input list 2: ", L2)
for i in range(max(len(L1), len(L2))):
  if i < len(L1):
    L3.append(L1[i])
  if i < len(L2):
    L3.append(L2[i])
print("Output: ", L3)
```

Output:

# output

```
Number of lines starting with 'A': 3

Number of lines starting with 'S': 5
```

# text.txt

```
Apple is a delicious fruit.
Banana is a yellow fruit.
Astronomy is the study of celestial objects.
Sunflowers are beautiful flowers.
Soccer is a popular sport.
Spring is my favorite season.
The sky is blue.
Sparrows chirp in the morning.
Autumn leaves are colorful.
Strawberries are sweet.
```

30. Write a user define function in python to display word in python to read lines from a text file and display those words which are more than five character’s.

Source Code:

```
def display_words_more_than_five_characters(file_path):
    # Open the file for reading using 'with' statement
    with open(file_path, 'r') as file:
        # Read all lines from the file
        lines = file.readlines()

        # Display words with more than five characters
        for line in lines:
            words = line.split()
            for word in words:
                if len(word) > 5:
                    print(word)

# Call the function with the file name
display_words_more_than_five_characters("text.txt")
```

Output:

```
# output
delicious
fruit.
Banana
yellow
fruit.
Astronomy
celestial
objects.
Sunflowers
beautiful
flowers.
Soccer
popular
sport.
Spring
favourite
season.
Sparrows
morning.
Autumn
leaves
```

# text.txt

```
Apple is a delicious fruit.
Banana is a yellow fruit.
Astronomy is the study of celestial objects.
Sunflowers are beautiful flowers.
Soccer is a popular sport.
Spring is my favorite season.
The sky is blue.
Sparrows chirp in the morning.
Autumn leaves are colorful.
Strawberries are sweet.
```