Cycle 1.3

1. Create a string from given string where first and last characters exchanged.

Algorithm.

① Start

② Enter a word

③ Swap last and first characters.

Program.

```
word = input("Enter a word : ")
first = word[0]
last = word[-1]
swap = last + word[1:-1] + first
print("Given word is :", word)
print("After swapping :", swap)
```

occupat

Enter a coord : cycle

Given coord is : cycle

After swapping : elycle

2. Accept the radius from user and find area of a circle.

Algorithm:

① Start
② Input radius (r)
③ PI = 3.14
④ area = PI * r * r
⑤ print area
⑥ Stop.

Program:

```
r = float(input("Enter radius of circle :"))
PI = 3.14
area = float(PI * r * r)
print("Area = ", area)
```

Output

Enter radius of circle : 4

Area = 50.24

3. Find biggest of 3 numbers entered.

Algorithm.

① Start

② Enter 3 numbers (n_1, n_2, n_3)

③ largest = 0

④ if $n_1 > n_2$ and $n_1 > n_3$

largest = n_1

else if ($n_2 > n_1$) and ($n_2 > n_3$)

largest = n_2

else largest = n_3

⑤ print largest

⑥ stop.

Program.

```
n1 = float(input("First number : "))
n2 = float(input("Second number : "))
n3 = float(input("Third number : "))
largest = 0
if (n1>n2) and (n1>n3):
        largest = n1
elif (n2>n1) and (n2>n3):
        largest = n2
else:
        largest = n3

Print("largest number = ", largest)
```

Output

First number 34

Second number 11

Third number 22

largest number = 34.0.

4. Accept a file name from user and print extension of that.

Algorithm.

① Start
② Input the file name (filename)
③ extension = filename.split(".")
④ print(extension[-1])
⑤ Stop.

Program.

```
filename = input("input file name")
f_extns = filename.split(".")
print("File name :", filename)
print("the extension is :", f_extns[-1])
```

Output

Input the file name : rose.pdf

File name : rose.pdf

the extension is : pdf

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

Algorithm.

① Start
② clr = []
③ Enter size of list
④ Enter colors
⑤ append colors to clr[]
⑥ print color list (clr[])
⑦ print clr[0]
⑧ print clr[-1]
⑨ stop.

program.

```
clr = []
n = int(input(" Enter total number of colors : "))
print(" Enter colors")
for i in range(0,n):
        item = input()
        clr.append(item)
print(" Given colors are :", clr)
print(" First color is :", clr[0])
print(" Last color is :", clr[-1])
```

Output

Enter total number of colors : 4

Enter colors

Red

Blue

Rose

Green

Given colors are : ['Red', 'Blue', 'Rose', 'Green']

First color is : Red

Last color is : Green.

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

Algorithm:

① Start
② Enter an integer(n)
③ result = n+nn+nnn
④ print result
⑤ Stop.

program:

```
n = int(input("Enter an integer (n) : "))
cal = int(n+n*n+n*n*n)
print("After calculation (n+nn+nnn) : ", cal)
```

Output

Enter an integer (n) : 2

After calculation (n+nn+nnn) : 14

7. print all colors from color-list1 not contained on color list2.

Algorithm:

① clr list1 = [], clr list2 = []
② Enter size of list1
③ Enter colors for list1
append colors to clr list1
④ enter size of list2
⑤ Enter colors for list2
append colors to clr list2
⑥ convert clr list1 to set (set1)
⑦ convert clr list2 to set (set2)
⑧ print (set1. difference(set2))

program:

```
clr list1 =[]
clr list2 =[]
n1 = int( input ("Enter size of list1 : "))
print ("Enter colors for list : \n")
for i in range (0, n1):
        elem1 = input()
        clr list1. append (elem1)
n2 = int (input ("Enter size of list2 : "))
print ("Enter colors for list2 : \n")
for i in range(0, n2):
        elem2 = input()
        clr list2. append (elem2)
```

```
set1 = set(clr_lst1)
set2 = set(clr_lst2)
print("color_lst1 =", clr_lst1)
print("color_lst2 =", clr_lst2)
print("color_lst1 - color_lst2 :", set1.difference(set2))
```

Output

```
Enter size of lst1 : 3
Enter colors for lst1 :
blue
green
red
Enter size of lst2 : 2
Enter colors for lst2 :
red
rose
color_lst1 = ['blue', 'green', 'red']
color_lst2 = ['red', 'rose']
color_lst1 - color_lst2 : {'blue', 'green'}
```

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

Algorithm

① Start
② enter first string (S_1)
③ enter second string (S_2)
④ def char_mix(a,b):
    new_a = b[0:1]+a[1:]
    new_b = a[0:1]+b[1:]
⑤ return new_a+" "+new_b
⑥ print (S_1, S_2)

Program

```
S1 = input("Enter first string :")
S2 = input("Enter second string :")
def char_mix(a,b):
    new_a = b[0:1]+a[1:]
    new_b = a[0:1]+b[1:]
    return new_a+" "+new_b
print(char_mix(S1, S2))
```

output

Enter first string : python

Enter second string : function.

Python Function.

9. Sort dictionary in ascending and descending order.

Algorithm.

① Start

② Input a dictionary

③ convert dictionary to list

④ Print in ascending order

lst.sort()

⑤ print in descending order

lst.sort(reverse = True)

⑥ Stop.

Program.

```
Dictionary = {'Mani':40, 'Ravi':30, 'Arun':50, 'Anupama':13,
              'Jibiya':10}
print("Dictionary :", Dictionary)
print("\n")
lt = list(Dictionary.items())
lt.sort()

print("Ascending order :", lt)
print("\n")
ds = list(Dictionary.items())
ds.sort(reverse = True)

print("Descending order :", ds)
```

Output

Dictionary : {'Manu' : 40 , 'Ravi' : 30, 'Arun' : 50,
'Anupama' : 13 , 'Jibiya' : 10}

Ascending order : [('Anupama', 13) , ('Arun', 50) , ('Jibiya', 10),
('Manu', 40) , ('Ravi', 30)]

Descending order : [('Ravi', 30) , ('Manu', 40) , ('Jibiya', 10),
('Arun', 50), ('Anupama', 13)]

10. Merge 1000 dictionaries.

Algorithm

① Start

② DICT1 = { }, DICT2 = { }

③ DICT3 = DICT1.copy()

④ DICT3.update(DICT2)

⑤ Print DICT3

⑥ Stop.

program

```
DICT1 = {'Manu' : 40, 'Ravi' : 30}
DICT2 = {'Arun' : 50, 'Anupama' : 13, 'Ishiya' : 10}
print ("Dictionary1 : ", DICT1)
print("Dictionary2 : ", DICT2)
DICT3 = DICT1.copy()
DICT3.update(DICT2)
print("After merging : ", DICT3)
```

output

Dictionary 1 : {'Manu' : 40, 'Ravi' : 30}

Dictionary 2 : {'Arun' : 50, 'Anupama' : 13, 'Jibiya' : 10}

After merging : {'Manu' : 40, 'Ravi' : 30, 'Arun' : 50,
'Anupama' : 13, 'Jibiya' : 10}

11. Find gcd of 1000 numbers.

Algorithm.

① Start
② Enter first number (n_1)
③ Enter second number (n_2)
④ gcd = math.gcd(n_1, n_2)
⑤ Print gcd
⑥ Stop

Program.

```
import math
n1 = int (input ("Enter first number : "))
n2 = int(input ("Enter second number : "))
gcd = math.gcd (n1, n2)
print ("gcd (", n1, " , " , n2, ") = " , gcd)
```

output

Enter first number : 7

Enter second number : 21

gcd (7, 21) = 7

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

Algorithm:

1. Start
2. integers = [], removeven = []
3. Enter size of list
4. Enter elements.

   append elements to integers[]
5. for i in range(0,n)

   if(item%2 != 0)

   append element to removeven[]
6. print removeven
7. Stop.

program.

```
integers = []
removeven = []
n = int(input("Enter size of list : "))
print("Enter integers : \n")
for i in range(0,n):
        item = int(input())
        integers. append (item)
        if (item %2!= 0):
                removeven . append (item)

print("Given list :", integers)
print("list removing even numbers :", removeven)
```

Output

Enter the size of list : 4

Enter integers :

1

2

3

4

Given list : [1, 2, 3, 4]

list removing even numbers : [1, 3]