

Cycle 1-3

Programs:

Aim: Create a String from given String where First and last characters exchanged [eg:- Python \rightarrow noython]

Algorithm

1. Start
2. Read a string.
3. Create a new string
 - 3.1 Store first character, and last character, and middle characters
 - 3.2. Add last + middle + first characters
4. Print the new string
5. Stop

Program

```
str = input ("Enter a string")  
new = str[-1:] + str[1:-1] + str[:1]  
print (new)
```

Output

Enter a string: noaths

earthos

Programs: 2

Aim: Accept the radius from user and
Find area of circle.

Algorithms

Steps

1. Start
2. Read radius From the user .
3. Define PI=3.14
4. Find the area using equation
5. Print area.
6. STOP.

program :

```
radius = float(input("Enter the radius"))
```

```
PI = 3.14
```

```
Area = PI * radius * radius
```

```
print ("Area of circle is ", Area)
```

Output

Enter the radius: 5

Area of Circle is 78.5

Program : 3

Aims: Find biggest of 3 numbers entered.

Algorithm

1. Start
2. Read First number.
3. Read Second number.
4. Read third number.
5. set largest equal to zero.
6. IF First number greater than Second and Third, set largest = First
 - * 6.1. elseif Second number greater than First and Third, Set largest = second
 - 6.2. else set largest = third.
7. print largest number.
8. Stop.

Program

```
num1 = float(input("First no:"))
num2 = float(input("Second no:"))
num3 = float(input("Third no:"))
```

largest = 0

If ($n_1 > n_2$) and ($n_1 > n_3$):

 largest = n_1

elif ($n_2 > n_1$) and ($n_2 > n_3$):

 largest = n_2

else:

 largest = n_3

```
Print("the largest number is",
      largest)
```

Output

First no: 15

Second no: 9

Third no: 10

The largest number is 15

Programs : 4

Algo: Accept a filename from user
and print extension of that.

Algorithm

1. Start
2. Read a file name.
3. split and store the File name
using split function
4. print the extension of file.
5. Stop.

Program

```
File = input ("Enter the Filename:")
```

```
f_extens = file.split (".")
```

```
Print ("extension of file is", f_extens[-1])
```

Output

Enter the filename : Sample.pdf

extension of file is pdf

Program : 5

Aim: Create a list of colors from color names entered by user. Display first and last color.

Algorithm

1. Start
2. read a list of color separated by comma
3. colors split using split function.
4. Print first color
5. Print last color
6. Stop

Program

```
list = input("Enter the color: ")
list_color = list.split(",")
print ("first Color:", list_color[0])
print ("last Color:", list_color[-1])
```

Output

Enter the color: red, yellow, green,

First color: red

last color: green.

Program : 6

Aim: Accept an integer n and compute
 $n + nn + nnn$ {
 }

Algorithm

1. Start
2. Read a number
3. Create a variable for store number
4. Create another variable for n^2
5. Create variable for n^3
6. Print sum of 3 variables.
7. Stop.

Program

```
n = int(input("Enter the number:"))

n1 = int(n)
n2 = int(n*n)
n3 = int(n*n*n)

print(n1+n2+n3)
```

Output

Enter the number: 5

155

Programs:

Aims: print out all colors from color-list1 not contained in color-list2.

Algorithm

1. Start
2. Create a list1 of colors
3. Create another list2 of colors.
4. Print the colors from list1
that not in list2 using difference
5. Stop.

Program

```
list1 = set(['white', 'red', 'black'])
```

```
list2 = set(['red', 'yellow', 'green'])
```

```
Print(list1.difference(list2))
```

Output

{'black', 'white'}

Program:8

Aim: Create a Single String Separated with Space From Two Strings by Swapping the character at position.

B. One.

Algorithm

1. Start
2. define a function character mix.
 - 2.1. store first character and of second string and characters of first string after 1st position
 - 2.2. store first character of 1st string and characters of second string after 1st position
 - 2.3. Return the new strings.
3. Read First string
4. Read Second string
5. print Call the function.
6. stop.

Program

```
def char_mix(x,y):  
    new_x=y[:1]+x[1:]  
    new_y=x[:1]+y[1:]  
    return new_x+new_y  
  
x=input("enter 1st string :")  
y=input("enter 2nd string :")  
print(char_mix(x,y))
```

Output

enter 1st string : maths

enter 2nd string : python

maths python

Program: 9.

Aim: Sort dictionary in ascending
and descending order

Algorithm

1. Start
2. Create a dictionary
3. Print the dictionary
4. Create a list to store items and
Sort the list using function
5. Print the dictionary in ascending order
6. Create a list to store items and
Sort the dictionary in descending
order
7. Print the descending list
8. Stop.

Program

```
dic = {'Ads': 40, 'WP': 54, 'PL': 56}
```

```
Print ("The dictionary is", dic)
```

```
Li1 = List (d.items ())
```

```
Li1.sort ()
```

```
Print ("List is ascending order:", Li1)
```

```
Li2 = List (d.items ())
```

```
Li2.sort (reverse = True)
```

```
Print ("List is descending order:", Li2)
```

Output

The dictionary is $\{ \text{'ads': 45}, \text{'cup': 54}, \text{'PL': 56} \}$

List in ascending order: $[(\text{'Ads'}, 45), (\text{'PL'}, 56), (\text{'cup'}, 54)]$

List in descending order: $[(\text{'cup'}, 54), (\text{'PL'}, 56), (\text{'Ads'}, 45)]$

Programs : 10

Aims: merge two dictionaries.

Algorithm

1. Define merge function
2. Enter dict 1 and dict 2
3. merge both to dict 3.
4. print the merged new dictionary
dict3

Program

```
def merge (dict1,dict2):  
    res= dict1|dict2
```

```
    return res.
```

```
dict1= {'x':10,'y':8}
```

```
dict2= {'a':5,'b':4}
```

```
dict3= merge(dict1,dict2)
```

```
print(dict3)
```

Output

{'x': 10, 'y': 8, 'a': 6, 'b': 4}

Programs: II

Aim: Find gcd of 2 numbers

Algorithm

1. start
2. get 2 integer inputs
3. use while loop to check if both numbers are divisible by any number(i) with out any remainder.
4. If true, then $\text{GCD} = i$
5. print GCD of two numbers.
6. stop.

Program

```
n1 = int(input("Enter 1st number:"))  
n2 = int(input("Enter 2nd number:"))
```

i=1

while (i < n1 and i < n2):

 if (n1 % i == 0 and n2 % i == 0):

 gcd = i

 i = i + 1

print("GCD is", gcd)

Output

Enter first number: 406

Enter Second number: 555

GCD is:

Programs: 12

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

Algorithm

1. Start
2. Create a list of integers.
3. Print the list
4. Use for loop to take each element
 - 4.1 Use if statement to take even numbers and remove from list.
5. Print the list.

6. Stop

Program

```
num = [11, 22, 25, 36, 43, 50]
```

```
Print ("original list: ")
```

```
Print (num)
```

```
for i in num:
```

```
    if (i % 2 == 0)
```

```
        num.remove (i)
```

```
Print ("list after remove even no: ")
```

```
Print (num)
```

Output

original list:

[11, 22, 25, 36, 43, 50]

list after remove even no:

[11, 25, 43]