Q.1 "**Write a program to display different data types**"

```
a = 11
b = 112232.09
c = 12j
d = "Abhishek"
print("The type of a is",type(a))
print("The type of b is",type(b))
print("The type of c is",type(c))
print("The type of d is",type(d))
```

```
The type of a is <class 'int'>
The type of b is <class 'float'>
The type of c is <class 'complex'>
The type of d is <class 'str'>
```

Q.2 "**Write a program to perform different arithmetic functions in python**"

```
a = int(input("Enter value of a: "))
b = int(input("Enter value of b: "))
print("Addition of  a and b", a+b)
print("Subtraction of  a and b",a-b)
print("Multipliation of  a and b",a*b)
print("Division of  a and b",a/b)
print("Modulus/Remainder of  a and b",a%b)
print("Floar division of  a and b",a//b)
print("Exponent of  a and b",a**b)
```

```
Enter value of a: 5
Enter value of b: 3
Addition of  a and b 8
Subtraction of  a and b 2
Multipliation of  a and b 15
Division of  a and b 1.6666666666666667
Modulus/Remainder of  a and b 2
Floar division of  a and b 1
Exponent of  a and b 125
```

Q.3 "**Write a program to print a calendar**"

```
import calendar
yy = int(input("Enter the required year before 2010: "))
mm = int(input("Enter the month:"))

print(calendar.month(yy,mm))
```

```
Enter the required year before 2010: 2008
Enter the month:12
   December 2008
Mo Tu We Th Fr Sa Su
```

```
     1  2  3  4  5  6  7
     8  9 10 11 12 13 14
    15 16 17 18 19 20 21
    22 23 24 25 26 27 28
    29 30 31
```

Q.4 **"Write a program to find maximum of two numbers"**

```
n = int(input("Enter the value for n: "))
d = int(input("Enter the value for d: "))

if (n>d):
  print("The maximum is n: ",n)
else:
  print("The maximum is d: ",d)
```

```
    Enter the value for n: 34
    Enter the value for d: 76
    The maximum is d:  76
```

Q.5 **"Write a program to find wether the input number is prime or not"**

```
num = int(input("Enter the number: "))
c = 0
for i in range (1,num+1):
  if(num%i == 0):
    c += 1

if (c==2):
  print("The number is prime", num)
else:
  print("The number is not prime", num)
```

```
    Enter the number: 13
    The number is prime 13
```

Q.6 **"Write a program to find factorial of a number"**

```
num = int(input("Enter the number: "))
fact = 1

for i in range(1,num+1):
  fact *= i

print(fact)
```

```
    Enter the number: 6
    720
```

Q.6 **"Write a program to print fibonacci series to nth term"**

```
n = int(input("Enter the nth term: "))
a = 0
b = 1
print(a)
print(b)

for i in range (2, n+1):
  c = a+b
  a = b
  b = c
  print(c)
```

```
Enter the nth term: 12
0
1
1
2
3
5
8
13
21
34
55
89
144
```

Q.7 **"Write a program to heck wether the number is a armstrong number or not"**

```
num_1 = int(input("Enter the number: "))
n = num_1
count=0
while(n>0):
    count=count+1
    n=n//10

n = num_1
d = 0
sum = 0
while (n>0):
  d = n%10
  sum += pow(d,count)
  n = n//10

if(sum == num_1):
  print("The number is Armstrong: ",num_1)
else:
  print("The number is not Armstrong!")
```

```
Enter the number: 153
3 0
153
The number is Armstrong:  153
```

Q.8 **"Write a python script to print the current date in the following format"**

"Sun Aug 7 11:46:23 IST 2021"

```
import time # importim=ng time package

ntime = time.localtime()
print(time.strftime("%a %b %d %H:%M:%S %Z %Y",ntime))

'''
%a is Abbreviated weekday name.
%b is Abbreviated month name.
%d is Day of the month as a decimal number [01,31].
%H is Hour (24-hour clock) as a decimal number [00,23].
%M is Minute as a decimal number [00,59].
%S is Second as decimal number [00, 61].
%Z is Time zone name (no characters if no time zone exists).
%Y is Year with century as a decimal number.  '''
```

```
Sat Aug 07 06:24:15 UTC 2021
'\n%a is Abbreviated weekday name.\n%b is Abbreviated month name.\n%d is Day of the
month as a decimal number [01,31].\n%H is Hour (24-hour clock) as a decimal number
[00,23].\n%M is Minute as a decimal number [00,59].\n%S is Second as decimal number
```

Q.9 **"Write a program to create a timer that will countdown in reverse order"**

```
import time

count_sec = 5   #you can input also for seconds
for i in reversed(range(count_sec+1)):
  if(i>0):
    print(i,end = '>>> ', flush = True)
    time.sleep(1)
  else:
    print("ReStart")
```

```
5>>> 4>>> 3>>> 2>>> 1>>> ReStart
```

Q.10 **"Write a program to input an array and reverse it"**

```
# number of elements
n = int(input("Enter number of elements : "))

# Below line read inputs from user using map() function
a = list(map(int,input("\nEnter the numbers : ").strip().split()))[:n]

print("\nList is - ", a)

a.reverse()
print(a)
```

```
Enter number of elements : 5

Enter the numbers : 2 1 45 23 4

List is -  [2, 1, 45, 23, 4]
[4, 23, 45, 1, 2]
```

Q.11 **"Write a program to sort the elements of a list"**

```
a = [12,34,2342,123,22,4,23,12] #list
a.sort()
print(a)
```

```
[4, 12, 12, 22, 23, 34, 123, 2342]
```

Q.12 **"Write a program to find the largest element in the list"**

```
a = [12,34,2342,123,22,4,23,12] #list
b = a[0]
print(len(a))
for i in range(0,len(a)):
  if(a[i]>b):
    b = a[i]

print(b)
```

```
8
2342
```

Q.13 "**Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays**.

Example 1:

```
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.
```

Example 2:

```
Input: nums1 = [1,2], nums2 = [3,4]
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5"
```

```
n1 = int(input("Enter the limit of first aaray: "))
n2 = int(input("Enter the limit of second array: "))
num1 = list(map(int,input("\nEnter the numbers of num1 array : ").strip().split()))[:n1]
```

```
num2 = list(map(int,input("\nEnter the numbers of num2 array : ").strip().split()))[:n2]
z=0
num3 = [None] * (n1 + n2)
for i in range (0,n1):
  num3[z] = num1[i]
  z = z+1

for j in range(0,n2):
  num3[z] = num2[j]
  z = z+1

num3.sort()
print("The merger sorted array: ",num3)

d = len(num3)
median = 0.0
if(d%2 == 0):
  f = d//2
  median = (num3[f] + num3[f-1])/2
else:
  f = (d-1)//2
  median = num3[f]

print("The median is: ",median)
```

```
Enter the limit of first aaray: 2
Enter the limit of second array: 2

Enter the numbers of num1 array : 1 3

Enter the numbers of num2 array : 2 4
The merger sorted array:  [1, 2, 3, 4]
The median is:  2.5
```

Q.14 "**The count-and-say sequence is a sequence of digit strings defined by the recursive formula**:

countAndSay(1) = "1" countAndSay(n) is the way you would "say" the digit string from countAndSay(n-1), which is then converted into a different digit string. To determine how you "say" a digit string, split it into the minimal number of groups so that each group is a contiguous section all of the same character. Then for each group, say the number of characters, then say the character. To convert the saying into a digit string, replace the counts with a number and concatenate every saying.

Example 1:

Input: n = 1 Output: "1" Explanation: This is the base case.

Example 2:

Input: n = 4 Output: "1211" Explanation: countAndSay(1) = "1" countAndSay(2) = say "1" = one 1 = "11" countAndSay(3) = say "11" = two 1's = "21" countAndSay(4) = say "21" = one 2 + one 1 = "12" + "11" = "1211" "

```
def countAndSay(curr):
    result = "" # to store the term after 'curr'
    i = 0       # to iterate over 'curr'
    # Need to iterate over 'curr', and also count the
    # number of digits that occur in the same group
    while i < len(curr):
        count = 1 # To store how many times a digit occured
        # Inner while loop compares current digit and the next digit
        while i + 1 < len(curr) and curr[i] == curr[i+1]:
            i += 1
            count += 1
        result += (str(count) + curr[i])
        i += 1
    return result

# Driver code
number = "1" # First member is always 1.
n = int(input("Enter the number: ")) # Number of members in the sequence
for i in range(n-1):
    number = countAndSay(number)

print(number)
```

```
Enter the number: 4
1211
```

PATTERN

Q.1 "**Write a program to print the pattern**"

```
# # # # # # #
# # # # # # #
# # # # # # #
# # # # # # #
# # # # # # #
```

```
# Algorithm
a ="#"

for i in range(5):
  for j in range (7):
    print(a, end=" ")

  print("\n")
```

```
# # # # # # #

# # # # # # #

# # # # # # #
# # # # # # #

# # # # # # #
```

Q.2 "**Write a program to print the pattern**"

```
1
2 3
4 5 6
7 8 9 10
..... to n term
```

```
# Algorithm
n = int(input("Enter the number: "))
a = 1
for i in range(i):
  for j in range(1,i+1):
    print(a, end = " ")
    a = a+1
  print("\n")
```

```
Enter the number: 4


1

2 3

4 5 6

7 8 9 10
```

Q.3 "**Write a program to print the pattern**"

```
1
22
333
4444
55555
......to nth term
```

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

for i in range(1,n+1):
  for j in range(i):
    print(i, end="")

  print("\n")
```

```
Enter the number: 5
1

22

333

4444

55555
```

Q.4 "**Write a program to print the pattern**"

```
from nth term....
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

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

for i in range(n,0,-1):
  for j in range (i):
    print(i,end = " ")

  print("\n")
```

```
Enter the number: 5
5 5 5 5 5

4 4 4 4

3 3 3

2 2

1
```

Q.5 "**Write a program to print the pattern**"

```
1
21
321
```

```
4321
54321
.....to nth term
```

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

for i in range(1,n+1):
  for j in range(i):
    print(i,end="")
    i = i-1

  print("\n")
```

```
Enter the number: 5
1

21

321

4321

54321
```

Q.6 **"Write a program to print the pattern"**

```
from nth term....
0 1 2 3 4 5
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0
```

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

for i in range(n,0,-1):
  for j in range (0,i+1):
    print(j,end= " ")

  print("\n")
```

```
Enter the number: 5
0 1 2 3 4 5

0 1 2 3 4
```

```
    0 1 2 3

    0 1 2

    0 1
```

Q.7 **"Write a program to print the pattern"**

```
1
121
12321
1234321
123454321
...... for n terms
```

```
#  Algorithm
n = int(input("Enter the number: "))
num = 0
for i in range(1,n+1):
  for j in range(i):
    num = num+1
    print(num,end ="")

  for z in range(i-1):
    num = num-1
    print(num, end = "")

  num = 0
  print("\n")
```

```
    Enter the number: 5
    1

    121

    12321

    1234321

    123454321
```

Q.8 **"Write a program to print the pattern"**

```
0
0 1
0 2 4
0 3 6 9
0 4 8 12 16
```

```
0 5 10 15 20 25
......upto n terms
```

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

for i in range(0,n+1):
  for j in range(0,i+1):
    print(i*j,end = " ")

  print("\n")
```

```
Enter the number: 6
0

0 1

0 2 4

0 3 6 9

0 4 8 12 16

0 5 10 15 20 25

0 6 12 18 24 30 36
```

Q.9 **"Write a program to print the pattern"**

```
    1
   12
  123
 1234
12345
....n terms
```

```
# Algorithm
n = int(input("Enter the number: "))
l=1

for i in range(n+1,0,-1):
  for z in range(1,i):
    print(" ",end="")

  for j in range(1,l):
    print(j,end="")

  l = l+1
  print("\n")
```

```
     Enter the number: 5


         1

        12

       123

      1234

     12345
```

Q.10 **"Write a program to print the pattern"**

```
.....from n terms
1 2 3 4 5 5 4 3 2 1
  1 2 3 4 4 3 2 1
    1 2 3 3 2 1
      1 2 2 1
        1 1
```

```
#Algorithm
n = int(input("Enter the limit: "))
z = 1
f = n
print("\n")
for i in range(1,n+1):
  for k in range(z):
    print(" ",end="")

  for j in range(1,f+1):
    print(j,end="")

  for l in range(f,0,-1):
    print(l,end="")

  z = z+1
  f = f-1
  print("\n")

# Output
```

```
     Enter the limit: 5


      1234554321
```

```
12344321

 123321

  1221

   11
```

TUPLE

Perfrom basic operations on tuples

```
tup = (1,2,3,4)
print("Printing element from index 1: ", tup[1:4])
print("Repeating the elements: ", tup*2)

# output
```

```
Printing element from index 1:  (2, 3, 4)
Repeating the elements:  (1, 2, 3, 4, 1, 2, 3, 4)
```

```
tup2 = ("sparta",'z', 5-4j)
print("Two tuples added: ", tup+tup2)
print("The minimum value in tup: ", min(tup))
print("The minimum value in tup: ", max(tup))
#output
```

```
Two tuples added:  (1, 2, 3, 4, 'sparta', 'z', (5-4j))
The minimum value in tup:  1
The minimum value in tup:  4
```

LIST

1. **Lists are mutable, with square braces**
2. **Tuples are immutable, with round braces**
3. **Dictionaries are mutable, with curly braces**

"Perform basic operations on lists"

```
# Operations

l1 = [1,2,3,4,"Abhi"]
print("Print a element at different indexes: ", l1[0],l1[3])
l1[0] = "hello"
print(l1)
l1.append("sparta")
```

```
print("Add an element: ",l1)
l1.pop(2)  # removing element from index 2
print("The element removed at 2nd index: ",l1)
#output
```

```
Print a element at different indexes:  1 4
['hello', 2, 3, 4, 'Abhi']
Add an element:  ['hello', 2, 3, 4, 'Abhi', 'sparta']
The element removed at 2nd index:  ['hello', 2, 4, 'Abhi', 'sparta']
```

```
# Operations

l2 = [3,45,23,212,10,35,234]
l2.reverse()
print("The reversed list: ", l2)
l2.sort()
print("The sorted list: ",l2)
l2.insert(2,"Abhishek") #appending at index 2
print(l2)

#Output
```

```
The reversed list:  [234, 35, 10, 212, 23, 45, 3]
The sorted list:  [3, 10, 23, 35, 45, 212, 234]
[3, 10, 'Abhishek', 23, 35, 45, 212, 234]
```

```
# Operations

l3 = ["apple", "papaya","kiwi","banana","orange"]
l3.sort()
print("The sorted list: ",l3)

print(l3*3)

print("added lists: ",l2+l3)
# Output
```

```
The sorted list:  ['apple', 'banana', 'kiwi', 'orange', 'papaya']
['apple', 'banana', 'kiwi', 'orange', 'papaya', 'apple', 'banana', 'kiwi', 'orange',
added lists:  [3, 10, 'Abhishek', 23, 35, 45, 212, 234, 'apple', 'banana', 'kiwi', '
```

DICTIONARY

Dictionaries syntax : {key:value,pair}

```
d1 = {"Apple":60, "Mango":120, "Guava": 70, "Banana": 300}
print(type(d1))

# extracting keys only
print("Printing the keys: ", d1.keys())

# extracting the values
# extracting the values
print("Printing the values: ",d1.values())
```

```
<class 'dict'>
Printing the keys:  dict_keys(['Apple', 'Mango', 'Guava', 'Banana'])
Printing the values:  dict_values([60, 120, 70, 300])
```

```
#modifying the values
d1["Mango"] = 200
print(d1)

d2 = {"hello":1000, "Abhishek":2000}
d1.update(d2)            #vice versa can happen
print("The updated dict: ",d1)

d1.pop("hello")
print(d1)

# Output
```

```
{'Apple': 60, 'Mango': 200, 'Guava': 70, 'Banana': 300, 'Abhishek': 2000}
The updated dict:  {'Apple': 60, 'Mango': 200, 'Guava': 70, 'Banana': 300, 'Abhishek
{'Abhishek': 2000, 'Apple': 60, 'Banana': 300, 'Guava': 70, 'Mango': 200}
```

```
d1.pop("Apple")  # popped out value
```

```
60
```

+ Code + Text

✓ 0s completed at 12:24