Factorial operation using recursive function:

```python
# Factorial operation using recursive function


def factorial(n):

 if (n<=1):
    return n
 else:
    return n * factorial(n-1)

print("The result is: ", factorial(int(input("Enter a number: "))))
```

input

```
Enter a number: 5
The result is:  120
```

Array right shifting:

```python
#Initialize and Displays array
arr = [0,1, 2, 3, 4, 5];
print("Original array: ", arr);
n = int(input("How many time you want to Shift? "));

#Rotate the given array by n times toward right
for i in range(0, n):
    #Stores the last element of array
    last = arr[len(arr)-1];

    for j in range(len(arr)-1, 0, -1):
        #Shift element of array by one
        arr[j] = arr[j-1];

    #Last element of the array will be added to the start of the array.
    arr[0] = last;

print("After rotation: ", arr);
```

input

```
Original array:  [0, 1, 2, 3, 4, 5]
How many time you want to Shift? 2
After rotation:  [4, 5, 0, 1, 2, 3]
```

Find the missing number:

```python
#Find the missing number

def missing_numbers(num_list):
    original_list = [x for x in range(num_list[0], num_list[-1] + 1)]
    num_list = set(num_list)
    return (list(num_list ^ set(original_list)))

print("Missing number:", missing_numbers([1,2,3,5]))

```

input

```
Missing number: [4]
```

Array absolute distinct count:

```python
#Absolute distict count

from collections import Counter
import numpy as np

sampleArray = [-5,-3,-1,0,3,6]
print("Sample array: ", sampleArray)
absoluteArray = np.absolute(sampleArray)
print("Absolute array: ", absoluteArray)

def countDistinct(absoluteArray):

    # counter method gives dictionary of elements in list with their corresponding frequency.
    # using keys() method of dictionary data structure we can count distinct values in array
    return len(Counter(absoluteArray).keys())

if __name__=="__main__":
    print ("Distinct count: ",countDistinct(absoluteArray))
```

input

```
Sample array:  [-5, -3, -1, 0, 3, 6]
Absolute array:  [5 3 1 0 3 6]
Distinct count:  5
```

Problem 1: Recursive function

```
16
17  def val_input(n):
18      return isinstance(n, int) and n in range(11)
19
20  def get_num():
21      try:
22          n = int(input("Please only enter whole numbers between 0 and 10: "))
23          while not val_input(n):
24              n = int(input("Please only enter whole numbers between 0 and 10: "))
25          return n
26      except ValueError:
27          print("Invalid entry, try again.")
28          return get_num()
29
30
31  def run_program():
32
33      call_func = input("Would you like to run the program? Y/N  ").lower()
34
```

input

```
The current value of n is: 5

The current value of n is: 6

The current value of n is: 7

Course Passed.
```

Triplet:

```
1   a = [10, 50, 5, 1]
2   b = [10, 2, 5, 1, 8, 20]
3   c = [12, 6]
4   d = [15, 30, -1, 'v', 5]
5   e = [4, 3, 1, 2, 6]
6
7   def tri_count(l_nums):
8       if len(l_nums) < 3:
9           return 0
10      for i in range(len(l_nums)):
11          if not isinstance(l_nums[i], int):
12              print(i)
13              return 0
14
15      l_nums.sort()
16
17      p = 0
18      while p in range(len(l_nums)-2):
19          q= p+1
```

input

```
[5, 1, 10, 50]
```