1) List Comprehension: -

```
   l = list()
    for i in range(x+1):
        for j in range(y+1):
            for k in range(z+1):
                if(i+j+k!=n):
                    l.append([i,j,k])
    print(l)
```

2)Finding the runner up score

```
 python(2)

    a = max(arr)

    c = arr.count(a)

    for i in range(c):
        arr.remove(a)

    print(max(arr))
```

3)Nested list

```
if __name__ == '__main__':
    dic = {}
    s = list()
    for _ in range(int(input())):
        name = input()
        score = float(input())
        if score in dic:
            dic[score].append(name)
        else:
            dic[score] = [name]
        if score not in s:
            s.append(score)

    m = min(s)
    s.remove(m)
    m1=min(s)
    dic[m1].sort()
    for i in dic[m1]:
        print(i)
```

4)Finding the percentage

```
if __name__ == '__main__':
    n = int(input())
    student_marks = {}
    for _ in range(n):
        name, *line = input().split()
        scores = list(map(float, line))
        student_marks[name] = scores
```

```
    query_name = input()

    s=0

    for i in student_marks[query_name]:
        s = s+i
    print("{0:.2f}".format(s/3))
```

5) List

```
if __name__ == '__main__':
    N = int(input())
    arr = []
    for i in range(N):
        s = input().split()
        for i in range(1,len(s)):
            s[i] = int(s[i])

        if s[0] == "append":
            arr.append(s[1])

        elif s[0] == "insert":
            arr.insert(s[1],s[2])
        elif s[0] == "remove":
            arr.remove(s[1])
        elif s[0]  == "pop":
            arr.pop()

        elif s[0] == "sort":
            arr.sort()
        elif s[0] == "reverse":
            arr.reverse()
        elif s[0] == "print":
            print(arr)
```

6)Tuple

```
if __name__ == '__main__':
    n = int(raw_input())
    integer_list = tuple(map(int, raw_input().split()))

    print(hash(integer_list))
```

7) Introduction to sets

```
def average(array):
    # your code goes here
    array = set(array)

    return sum(array)/len(array)
```

```
if __name__ == '__main__':
    n = int(input())
    arr = list(map(int, input().split()))
    result = average(arr)
    print(result)
```

8)No Idea

```
n,m = map(int,input().split())
array = list(map(int,input().split()))
A=set(map(int,input().split()))
B=set(map(int,input().split()))
print(sum((i in A)-(i in B) for i in array))
```

9)set.add()

```
n = int(input())
set1=set()
for i in range(n):
    set1.add(input())
print(len(set1))
```

10)Introduction to sets

```
def average(array):
    # your code goes here
    array = set(array)
    return sum(array)/len(array)
if __name__ == '__main__':
    n = int(input())
    arr = list(map(int, input().split()))
    result = average(arr)
    print(result)
```

11)Set .discard(), .remove() & .pop()

```
n = int(input())
s = set(map(int,input().split()))
m = int(input())
for i in range(m):
    s1=list(input().split())
    if s1[0]=="pop":
        s.pop()
    elif s1[0]=="remove":
        s.remove(int(s1[1]))
    elif s1[0]=="discard":
        s.discard(int(s1[1]))
sum=0
for i in s:
    sum=sum+1
print(sum)
```

12)set.union()

```
n = int(input())
n1 = set(map(int,input().split()))
m=int(input())
m1=set(map(int,input().split()))
ans=n1.union(m1)
count=0
for i in ans:
    count=count+1
print(count)
```

13)set.intersection()

```
n = int(input())
n1 = set(map(int,input().split()))
m = int(input())
m1=set(map(int,input().split()))
ans=n1.intersection(m1)
count=0
for i in ans:
    count=count+1
print(count)
```

14)set.difference()

```
eng = int(input())
eng_roll = set(map(int,input().split()))

french = int(input())
french_roll = set(map(int,input().split()))
print(len(eng_roll.difference(french_roll)))
```

15)set.symmetric_difference() operation

```
n1 = int(input())
storage1 = set(input().split())
n2 = int(input())
storage2 = set(input().split())
storage3 = storage1.symmetric_difference(storage2)
print(len(storage3))
```

16) set.mutations()

```
def updateit(setA,s,command):
    if command == "update":
        setA.update(s)
    elif command == "difference_update":
        setA.difference_update(s)

    elif command == "intersection_update":
        setA.intersection_update(s)
```

```
    else:
        setA.symmetric_difference_update(s)
    return setA

a = int(input())
setA = set(map(int,input().split()))

for i in range(int(input())):
    command,len_of_set = input().split()
    s = set(map(int,input().split()))
    setA = updateit(setA,s,command)
print(sum(setA))
```

17) The captain's room

```
k = int(input())
rooms = list(map(int,input().split()))
a = set()
b = set()

for room in rooms:
    if room not in a:
        a.add(room)
        b.add(room)
    else:
        b.discard(room)
b = list(b)
print(b[0])
```

18)check subset

```
T = int(input())

for i in range(T):
    a = input()
    A = set(input().split())
    b = int(input())
    B = set(input().split())
    print(A.issubset(B))
```

19)Check Strict superset()

```
l = set(map(int,input().split()))
n = int(input())
val = ""
for i in range(n):
    l1 = set(map(int,input().split()))
    if len(l1.difference(l))==0:
        val = "True"
    else:
        val = "False"
        break
print(val)
```