Python 1 to 10

#2

```python
a=float(input("Enter 1st number"))
b=float(input("Enter 2nd nuber"))
c=((a**2)-(b**2))**0.5
print("Distance:",c)
```

#3

```python
import sys
a=int(sys.argv[1])
b=int(sys.argv[2])
c=a+b
print(c)
```

#4

```python
a=int(input("Enter the 1st number:"))
b=int(input("Enter the 2nd number"))
print("Number before swap is",a,b)
a=a+b
b=a-b
a=a-b
print("Number after swap:",a,b)
```

#5

```python
a=int(input("Enter the number:"))
if a%2==0:
    print("Given number",a,"is an even number")
else:
    print("Given number",a,"is an odd number")
if a == 2:
    print("Given number",a,"is an prime number")
if a >= 2:
    for i in range (2,(a//2)):
        if a % i == 0:
            print("Given number is not a prime number")
            break
        else:
            print("Given number is a prime number")
            break
```

#6

```python
a=int(input("Enter the range of list"))
ls = []
pl = []
nl = []
for i in range (a):
    b=int(input("Enter the number:"))
```

```python
        ls.append(b)
print(len(ls))
```

#7

```python
a = int(input("Enter the range of the list: "))
ls = []
for i in range(a):
    b = int(input("Enter the number: "))
    ls.append(b)
print("Your list is:", ls)

ls.sort()
n = len(ls)
if n % 2 == 0:
    # For even-length lists, calculate the average of the two middle
elements
    c = (ls[n // 2 - 1] + ls[n // 2]) / 2
    print("The median =", c)
else:
    # For odd-length lists, simply take the middle element
    c = ls[n // 2]
    print("Median =", c)
```

#9

```python
def lis(lst):
    r = []
    for i in lst:
        s = set()
        s.add(i)
        r.append(s)
    return(r)
lst = []
a = int(input("Enter the range of set:"))
for i in range (a):
    b = int(input("Enter the number: "))
    lst.append(b)
print(lis(lst))
```

#10

```python
a = int(input("Enter the range of list:"))
num = []
for i in range (a):
    b = int(input("Enter the number:"))
    num.append(b)
cubes = list(map(lambda x: x ** 3, num))
print("Your list elements:",num)
print("Cube of list elements:", cubes)
```