WordCount
```
from collections import Counter
s = input("Enter text: ").split()
for k, v in Counter(s).items():
    print(f"{k} - {v}")
```

FM Algorithm
```
import hashlib
CUSTOM_HASHES={1:(6,1,5),2:(11,7,5)}
def linear_hash(x,a,b,p):return (a*x+b)%p
def tz(x):return (x & -x).bit_length()-1 if x else 0
def fm(data,a,b,p):
    m=0;print("\nElement-wise computation:")
    for e in data:
        x=int(e) if e.isdigit() else int(hashlib.md5(e.encode()).hexdigest(),16)%p
        h=linear_hash(x,a,b,p);r=tz(h)
        print(f"Element: {e} | Hash: {h} | Binary: {format(h,'03b')} | r = {r}")
        m=max(m,r)
    return 2**m
elems=[e.strip() for e in input("Enter comma-separated stream elements: ").split(",") if e.strip()]
print("\nAvailable hash functions:")
[print(f"{i}: h(x)=({a}*x+{b}) mod {p}") for i,(a,b,p) in CUSTOM_HASHES.items()]
hf=int(input("\nChoose hash function number: ") or 1)
a,b,p=CUSTOM_HASHES.get(hf,(6,1,5))
print(f"\nEstimated number of distinct elements using hash {hf}: {fm(elems,a,b,p)}")
```

Bloom Filter
```
import math
m=int(input("Enter m: ").strip())
bits=[0]*m
f1=lambda x: x%m
f2=lambda x: (2*x+6)%m
ins=input("Insert ints (space, blank none): ").strip()
for x in map(int,ins.split()) if ins else []: bits[f1(x)]=bits[f2(x)]=1
qry=input("Check ints (space, blank none): ").strip()
for x in map(int,qry.split()) if qry else []:
    print(x,"->","probably present" if bits[f1(x)] and bits[f2(x)] else "not present")
print("Bit array:",list(enumerate(bits)))
```

Matrix Multiplication
```
import numpy as np
r1,c1=map(int,input("Enter rows,cols of A: ").split())
r2,c2=map(int,input("Enter rows,cols of B: ").split())
if c1!=r2:print("Matrix multiplication not possible")
else:
    A=np.array([list(map(int,input(f"A row {i+1}: ").split())) for i in range(r1)])
    B=np.array([list(map(int,input(f"B row {i+1}: ").split())) for i in range(r2)])
    print("Result:\n",A@B)
```

CRUD Operation

# Show all databases

```
show dbs
```

# Create or switch to a new database

```
use newClg
```

# Create a collection

```
db.createCollection("Atharva")
```

# Show all collections

```
show collections
```

# Insert a single document

```
db.Atharva.insert({name:"Atharva", class:"BE CS", place:"Thane", mobno:9876543210})
```

# Insert multiple documents

```
db.Atharva.insertMany([
  {name:"Darsh", class:"BE CS", place:"Pune", mobno:9999999999},
  {name:"Kunal", class:"BE CS", place:"Thane", mobno:8888888888}
])
```

# Retrieve all documents in readable format

```
db.Atharva.find().pretty()
```

# Retrieve documents with specific condition

```
db.Atharva.find({place:"Thane"}).pretty()
```

# Update all matching entries

```
db.Atharva.update(
  { class: "BE CS" },
  { $set: { mobno: 8888888888888 } },
  { multi: true }
)

# View updated data
db.Atharva.find().pretty()

# Drop (Delete) a Record Delete a single document
db.Atharva.deleteOne({ name: "Atharva" })

# Delete multiple documents
db.Atharva.deleteMany({ class: "BE CS" })

# Delete all records (empty the collection)
db.Atharva.deleteMany({})

# Drop (delete) a collection
db.Atharva.drop()

# Drop database
db.dropDatabase()
```