**1.Number system**

**Code:**

x = int(input("Enter number in base 10: "))

s = ''

*while* x!=1:

s+=str(x%2)

x = x//2

s += '1'

print(s[::-1])

**Output:**

Enter number in base 10: 73

1001001

**2.Removal of an element from a list**

**Code:**

def remove\_element(arr,rem):

l = []

for i in range(len(arr)):

if arr[i] != rem:

l.append(arr[i])

return l

l = [int(i) for i in input("Enter spaced integers to form a list: ").split()]

rem = int(input("Enter the element that is to be removed: "))

print(remove\_element(l,rem))

**Output:**

Enter spaced integers to form a list: 1 2 3 3 3 3 4 4 4 4 4

Enter the element that is to be removed: 4

[1, 2, 3, 3, 3, 3]

**3.Linearly searching for an element in a list**

**Code:**

def linear\_search(l,x):

count = 0

for i in l:

if i == x:

count+=1

return count if count else "Element not in list"

l = [int(i) for i in input("Enter spaced integers to form the list: ").split()]

x = int(input("Enter the element you want to search for: "))

print(linear\_search(l,x))

**Output:**

Enter spaced integers to form the list: 1 2 3 4 4 4 4 3 3 3 2 2 2

Enter the element you want to search for: 2

4

Enter spaced integers to form the list: 1 2 3 3 3 3 3

Enter the element you want to search for: 4

Element not in list

**4.Binary Search**

**Code:**

lower = 0

upper = len(l)

list1 = [2,3,4,5,10,15,21,29,90,91,92,107,204]

search = int(input("Enter the number to be searched for: "))

l = len(list1)//2

while True:

if list1[l]==search:

print(f'{search} found in position {l}')

break

elif list1[l] < search: l += l//2

else: l -= l//2

**Output:**

Enter the number to be searched for: 10

10 found in position 4

**5.Stacks**

**Code :**

def push(s, e):

s.append(e)

return s, e

def pop(s):

if len(s):

e = s.pop()

return s, e

else:

return s, "Stack Underflow"

def peek(s):

if len(s):

return s[-1]

else:

return "Stack Underflow"

def display(s):

for i in range(len(s)-1, -1, -1):

print(s[i])

return s

def Menu():

print("1. Push")

print("2. Pop")

print("3. Peek")

print("4. Display")

print("5. Exit")

stack = []

if \_\_name\_\_ == "\_\_main\_\_":

Menu()

operation = input("Enter your choice: ")

while operation != "5":

if operation == "1":

element = input("Element you want to push: ")

print(push(stack, element))

elif operation == "2":

print(pop(stack))

elif operation == "3":

print(peek(stack))

elif operation == "4":

print(display(stack))

else:

print("Enter a valid operation")

operation = input("Enter your choice: ")

**Output:**

1. Push

2. Pop

3. Peek

4. Display

5. Exit

Enter your choice: 1

Element you want to push: 3

(['3'], '3')

Enter your choice: 1

Element you want to push: 4

(['3', '4'], '4')

Enter your choice: 2

(['3'], '4')

Enter your choice: 1

Element you want to push: 4

(['3', '4'], '4')

Enter your choice: 4

4

3

['3', '4']

Enter your choice: 3

4

Enter your choice: 5

**6.Queues**

**Code :**

def enqueue(q, e):

q.append(e)

return q, e

def dequeue(q):

if len(q):

e = q.pop(0)

return q, e

else:

return q, "Empty Queue"

def peek(q):

if len(q):

return q[0]

else:

return "Empty Queue"

def display(q):

return ' '.join(q)

def Menu():

print("1. Enqueue")

print("2. Dequeue")

print("3. Peek")

print("4. Display")

print("5. Exit")

Menu()

queue = []

operation = input("Enter your choice: ")

while operation != "5":

if operation == "1":

element = input("Enter the element you want to enqueue: ")

print(enqueue(queue, element))

elif operation == "2":

print(dequeue(queue))

elif operation == '3':

print(peek(queue))

elif operation == "4":

print(display(queue))

else:

print("Enter a valid option")

Menu()

operation = input("Enter your choice: ")

**7.Bubble Sort**

**Code:**

arr = [1,3,2,1,2,4,-1,-1,-234,73,0,0,12,4,0,5]

*for* i *in* range(len(arr)):

*for* j *in* range(len(arr)-1):

*if* arr[j]>arr[j+1]:

arr[j],arr[j+1] = arr[j+1],arr[j]

print(arr)

**Output:**

[-234, -1, -1, 0, 0, 0, 1, 1, 2, 2, 3, 4, 4, 5, 12, 73]

**8.Insertion Sort**

**Code:**

arr = [1,3,2,1,2,4,-1,-1,-234,73,0,0,12,4,0,5]

*for* i *in* range(1,len(arr)):

key = arr[i]

j = i-1

*while* j>=0 and key < arr[j]:

arr[j+1] = arr[j]

j-=1

arr[j+1] = key

print(arr)

**Output:**

[-234, -1, -1, 0, 0, 0, 1, 1, 2, 2, 3, 4, 4, 5, 12, 73]

**9.Selection Sort**

**Code:**

*for* i *in* range(len(l)):

min\_idx = i

*for* j *in* range(i+1, len(l)):

*if* l[min\_idx] > l[j]:

min\_idx = j

l[i], l[min\_idx] = l[min\_idx], l[i]

print(l)

**Output:**

[-73, 0, 3, 3, 3, 5, 5, 7, 8, 34, 73, 234, 345, 456, 1234]

**10.Merge Sort**

**Code:**

l = [5,3,5,7,8,3,3,34,345,456,73,-73,0,234,1234]

l1 = [10,20,30,40]

l2= [30,70,120,190]

l = []

j = 0

*for* i *in* range(len(l1)):

*while* l2[j]<l1[i]:

l.append(l2[j])

j+=1

l.append(l1[i])

l.extend(l2[j:])

print(l)

**Output:**

[10, 20, 30, 30, 40, 70, 120, 190]

**11.File Handling -txt –removing all empty lines**

**Code:**

*with* open('abc.txt','r') *as* f:

s = f.read().split('\n')

*while* '' in s:

s.remove('')

*for* i *in* range(len(s)):

s[i]+='\n'

*with* open('abc.txt','w') *as* f:

f.writelines(s)

**Output:**

**Initially:**

something

nothing

everything

**Finally:**

something

nothing

everything

**12.File Handline -dat -reading**

**Code:**

*import* pickle

*with* open('abc.dat','rb') *as* f:

*try*:

*while* True:

s = pickle.load(f)

print(s)

*except* EOFError:

print("End of file reached")

**Output:**

bye

hello

how are you?

where are you?

bye

End of file reached

**13.File Handling -dat -writing**

**Code:**

*import* pickle

*with* open('abc.dat','wb') *as* f:

arr = ['something','nothing','everything']

pickle.dump(arr,f)

**14.File Handling -csv -reading**

**Code:**

*import* csv

*with* open('abc.csv','r') *as* f:

reader = csv.reader(f,*delimiter* = ',')

*for* row *in* reader:

*for* item *in* row:

print(item,*end* = ' ')

print()

**Output:**

Name Class

Someone 12

Nobody 11

**15.File Handling -csv -writing**

**Code:**

*import* csv

*with* open('abc.csv', 'w') *as* f:

writer = csv.writer(f)

writer.writerow(['Name', 'Class'])

writer.writerows([['Someone', 12], ['Nobody', 11]])

**16.Random alpha numeric string generator**

**Code:**

*import* random

numbers = range(11)

string = 'testString'

alpha\_numeric = ''

*for* i *in* range(random.randint(0,20)):

alpha\_numeric += str(random.choice(numbers))

*if* random.choice([True,False]):

alpha\_numeric += random.choice(string)

print(alpha\_numeric)

**Output:**

6S3g5n10g

**17.String Reversal**

**Code:**

x = input("Enter a string: ")

s = ''

*for* i *in* range(len(x)-1,-1,-1):

s += x[i]

print(s)

**Output:**

Enter a string: Hello

olleH

**18.Recursion**

**Code:**

def pow(*a*,*n*):

*if* *n* == 0:

*return* 1

*else*:

*return* *a*\*pow(*a*,*n*-1)

print(pow(2,10))

**Output:**

1024

**19.List Handling**

**Code:**

l = [1,2,3,4,5,6,7,7,87,7,7,7,7,7,7,73,3,3,3]

*while* 7 in l:

l.remove(7)

*for* i *in* range(len(l)):

l.append(i)

l.extend([4321,1234,5432,2345,6543,3456])

*del*(l[0])

l.insert(4,5)

print(l)

**Output:**

[2, 3, 4, 5, 5, 6, 87, 73, 3, 3, 3, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 4321, 1234, 5432, 2345, 6543, 3456]

**20.SQL Queries**

**Creating a table:**

create table something(

-> name text not null,

-> dept text not null);

**Inserting values in a table:**

insert into something values('Nobody','IT');

**Deleting values in a table:**

delete from something where name = 'Nobody';

**Deleting a table:**

drop table something;