In [12]:
```
l=[10,45,12,66,1,3,7,46]

for i in range(len(l)):
    for j in range(0,len(l)-i-1):
        if l[j]>=l[j+1]:
            l[j],l[j+1]=l[j+1],l[j]
        else:
            l[j]=l[j]
print(l)
```

executed in 8ms, finished 15:10:19 2024-08-20

```
[1, 3, 7, 10, 12, 45, 46, 66]
```

40. input --> "silent"

output --> "eilnst"

In [20]:
```
s=input("Enter the string:")

l=[]

for i in s:
    l.append(ord(i))


for i in range(len(l)):
    for j in range(0,len(l)-i-1):
        if l[j]>=l[j+1]:
            l[j],l[j+1]=l[j+1],l[j]
        else:
            l[j]=l[j]

char=[]

for i in l:
    char.append(chr(i))

print("".join(char))
```

executed in 4.17s, finished 15:16:14 2024-08-20

```
Enter the string:string
ginrst
```

In [25]:
```
s="The quick brown fox jumps over the lazy dog"
l=s.split()

for i in range(len(l)):
    for j in range(i+1):
        print(l[j],end=" ")
    print()
```

executed in 8ms, finished 15:19:57 2024-08-20

```
The
The quick
The quick brown
The quick brown fox
The quick brown fox jumps
The quick brown fox jumps over
The quick brown fox jumps over the
The quick brown fox jumps over the lazy
The quick brown fox jumps over the lazy dog
```

Sets

- Set is also one of the predefined class and treated as set data type.
- set is also used to store multiple values in it.
- Set does not allow duplicate values.
- set does not support indexing and slicing.
- set is both mutable and immutable , immutable in case of item assignment and mutable in case of add().
- Set does not allow mutable elements as its elements.

In [39]:
```
s1={1,2,3,4,5,"sham",(1,2,3)}
print(s1)
```

executed in 7ms, finished 15:32:01 2024-08-20

```
{1, 2, 3, 4, 5, 'sham', (1, 2, 3)}
```

In [40]:
```
s2={}
print(s2,type(s2))
print("="*50)
s3=set()
print(s3,type(s3))
```

executed in 7ms, finished 15:33:03 2024-08-20

```
{} <class 'dict'>
==================================================
<__main__.set object at 0x000001B74A11C650> <class '__main__.set'>
```

Inbuilt Functions

1. add()

In [41]:
```
s1={1,2,3,4,5,"sham",(1,2,3)}
s1.add(100)
print(s1)
```

executed in 8ms, finished 15:34:26 2024-08-20

```
{1, 2, 3, 4, 5, 'sham', 100, (1, 2, 3)}
```

2.remove()

In [44]:
```
s1={1,2,3,4,5,"sham",(1,2,3)}

s1.remove(4)
print(s1)
```

executed in 7ms, finished 15:35:17 2024-08-20

```
{1, 2, 3, 5, 'sham', (1, 2, 3)}
```

In [45]:
```
s1.remove(4)
```

executed in 28ms, finished 15:35:41 2024-08-20

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[45], line 1
----> 1 s1.remove(4)

KeyError: 4
```

3. discard()

In [60]:
```
t=(1,2,3,4)
s1={1,2,3,4,5,"sham",t}

s1.discard(t[2])
print(s1)
```

executed in 9ms, finished 15:41:46 2024-08-20

```
{1, 2, 4, 5, 'sham', (1, 2, 3, 4)}
```

In [47]:
```
s1.discard(4)
```

executed in 5ms, finished 15:36:29 2024-08-20

4.pop()

In [48]:
```
s1={1,2,3,4,5,"sham",(1,2,3)}

s1.pop()
print(s1)
```

executed in 8ms, finished 15:38:19 2024-08-20

```
{2, 3, 4, 5, 'sham', (1, 2, 3)}
```

In [69]:
```
s2={19,7,1,1.6,"hello","bye",5-6j,"string",False}

print(s2.pop())
print(s2.pop())
```

executed in 7ms, finished 15:42:45 2024-08-20

```
False
1
```

5. isdisjoint()

- checks for the null intersection

In [71]:
```
s1={1,2,3,4,5}
s2={4,6,7,8,9,10}

print(s1.isdisjoint(s2))
```

executed in 8ms, finished 15:44:32 2024-08-20

```
False
```

6. issubset()

In [81]:
```
s1={1,2,3,4,5}
s2={4,6,7,8,9,10}
s3={2,3,4}

print(s1.issubset(s2))
print(s3.issubset(s1))
```

executed in 8ms, finished 15:50:30 2024-08-20

```
False
True
```

7. superset()

In [91]:
```
s1={1,2,3,4,5}
s2={4,6,7,8,9,10}
s3={2,3,4}

print(s1.issuperset(s3))
```

executed in 8ms, finished 15:53:34 2024-08-20

```
True
```

8. union()

In [92]:
```
s1={1,2,3,4,5}
s2={4,6,7,8,9,10}

print(s1.union(s2))
```

executed in 8ms, finished 15:54:36 2024-08-20

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

9. intersection()

In [93]:
```
s1={1,2,3,6,4,5}
s2={4,6,7,8,9,10}

print(s1.intersection(s2))
```

executed in 7ms, finished 15:55:24 2024-08-20

```
{4, 6}
```

10. difference()

In [94]:
```
s1={1,2,3,6,4,5}
s2={4,6,7,8,9,10}

print(s1.difference(s2))
```

executed in 8ms, finished 15:56:29 2024-08-20

```
{1, 2, 3, 5}
```

11. symmetric difference()

In [96]:
```
s1={1,2,3,6,4,5}
s2={4,6,7,8,9,10}

print(s1.symmetric_difference(s2))
```

executed in 7ms, finished 15:58:27 2024-08-20

```
{1, 2, 3, 5, 7, 8, 9, 10}
```

46.write a program to print True if any common element in given two sets else print False

---

47.Write a program to print common elements from given three sets

---

48.input = [[1, 2, 2, 4, 3, 6],[5, 1, 3, 4],[9, 5, 7, 1],[2, 4, 1, 3,8]]

output = {1, 2, 3, 4, 5, 6, 7, 8, 9}

45. Write a program to find the frequency of characters in a given string

input ---> dedication

```
output ---> d-2
            e-1
            i-2
            c-1
            a-1
            t-1
            o-1
            n-1
```

53.The Uppal Cricket Stadium has to following rates for different types of seats:

```
Ordinary - 2500
Pavillion - 3500
Upper Pavillion - 4500
Commentary Box -6000
VIP - 8000
```

They are Giving 10% discount for online booking and 8% discount for advance booking and no discount is given for booking on match day from ticket window.

Enter the name, booking type like online, advance,or window booking.

Select the types of seats.

Compute the amount and print the ticket with proper format.

- Name
- Seat Type

- Seat Number (Seat Type first Character + random number between (1,5000))
- Booking id (first three characters of btype + Name first three characters + @uppal + 2024)
- Booking Type
- Discount Amount
- Booking Amount.

In [ ]: