List entering values, append, reading values from list and display

age=[]  
i=**1**limit=**5**print(**"entering values to the list"**)  
**while** i<=limit:  
 *#a=input("enter the age you want") # when we not give type"int()" the value stored as string value then, when we print  
 # - the list it will disply the value inside quotes like this-['56', '23', '56', '01', '2']-* a= int(input(**"enter age"**))  
 age.append(a)# append or add items to the list named age   
 i=i+**1***# print(i)*age.append(**"sruthi"**) *# in list we can add any datatype values*

print(age)# print the list age  
  
print(**"reading values from list"**)  
j=**0**sum=**0  
while** j<len(age):  
 sum= sum+j *#sum of ages in the list* print(age[j])  
 j=j+**1**print(**'age sum'**,sum)  
print(**'avg= '**,sum/len(age))*# average*output

entering values to the list

enter age>? 24

enter age>? 23

enter age>? 23

enter age>? 23

enter age>? 23

[24, 23, 23, 23, 23, 'sruthi']

reading values from list

24

23

23

23

23

sruthi

age sum 15

avg= 2.5

list check

list=[**50**,**53**,**500**,**5**,**96**,**100**]  
x=int(input(**"enter the number you want to check in the list"**))  
**for** i **in** list:  
 **if** x==i:  
 print(**"the number is in the list"**)  
 print (i)  
 **break  
else** :  
 print(**"the number you entered is not exist in the list"**)  
*# print(len(list))  
# print(list[4:])  
# print(list[0:5:2]) # print list alternate values# gap `2  
# print(list[::3])  
# print(list[::-1])# reverse indexing -1///////////// important*

output

**enter the number you want to check in the list>? 50**

the number is in the list

50

**Password check list**

listun=[**"abc1"**,**"abc2"**,**'abc3'**] *#u*listpwd=[**"p1"**,**"p12"**,**'p123'**] *#index*un=input(**"enter username:"**)  
index=**0  
for** u **in** listun:  
 **if** un==u:  
 pwd = input(**"enter password:"**)  
 **if** pwd==listpwd[index]:  
 *#x=listun.index(un) list name.index(...) # to find the position in list* print(index)  
 print(**'welcome'**,un)  
 **else**:  
 print(**'password incorrect'**)  
 **break** index=index+**1  
else** :  
 print(**"invalid user name"**)

output

enter username:>? abc1

enter password:>? p1

0

welcome abc1

**SET**

You can create a set using curly braces {} or the set() constructor.

Unordered You cannot change individual elements because sets are unordered collections.

Mutable Sets are mutable in the sense that you can add, remove, and update elements.

Un-indexed

No-duplicte allowed automatically handle duplicate elements, ensuring that all elements are unique.

*# Using curly braces*my\_set = {**1**, **2**, **3**, **4**, **5**,**1**,**3**,**3**,**2**}  
print(my\_set)  
>>>>>>>>{1, 2, 3, 4, 5}

*# OR Using set() constructor*set\_1 = set([**11**, **2**, **3**, **4**, **6**,**4**,**4**,**4**,**4**,**4**])  
print(set\_1)

>>>>>>>>>>>>>>>>>>>{2, 3, 4, 6, 11}

 **Adding Elements**

add() method to add an element to a set.

my\_set = {1, 2, 3}

my\_set.add(4)

print(my\_set) # Output: {1, 2, 3, 4}

**Removing Elements**

Use the remove() method to remove a specific element. This raises a KeyError if the element is not found. You can also use discard() which does not raise an error if the element is not found.

my\_set = {1, 2, 3, 4}

my\_set.remove(3)

print(my\_set) # Output: {1, 2, 4}

my\_set.discard(5) # Does nothing if 5 is not in the set

**Clearing a Set**

Use the clear() method to remove all elements from a set.

my\_set = {1, 2, 3}

my\_set.clear()

print(my\_set) # Output: set()

**Checking Membership**

Use the in keyword to check if an element is in a set.

my\_set = {1, 2, 3}

print(2 in my\_set) # Output: True

print(4 in my\_set) # Output: False

**Set Operations**

Sets support mathematical operations like union, intersection, and difference.

set1 = {1, 2, 3}

set2 = {3, 4, 5}

# Union

union\_set = set1 | set2

print(union\_set) # Output: {1, 2, 3, 4, 5}

# Intersection

intersection\_set = set1 & set2

print(intersection\_set) # Output: {3}

# Difference

difference\_set = set1 - set2

print(difference\_set) # Output: {1, 2}

# Symmetric Difference - returns a set containing elements that are in either of the sets but not in both.

sym\_diff\_set = set1 ^ set2

print(sym\_diff\_set) # Output: {1, 2, 4, 5}

*# convert list string to list integer*nums=[**'22'**,**'34'**,**'35'**,**'55'**]  
x=([int(i) **for** i **in** nums])  
print(type(nums[**0**]))  
print(type(x[**1**]))