# USE CASES  
  
  
# 1. ARITHMETIC OPERATOR USE CASE (SIMPLE INTEREST CALCULATOR)  
  
  
def simple\_interest(p, t, r):  
 print('The principal is', p)  
 print('The time period is', t)  
 print('The rate of interest is', r)  
  
 si = (p \* t \* r) / 100  
  
 print('The Simple Interest is', si)  
 return si  
  
  
simple\_interest(80000, 5, 8)  
  
# 2. DIVISIBILTY RULE  
  
a = 24  
  
if not (a % 2 == 0 or a % 3 == 0):  
 print(a, " is not divisible by either 2 or 3")  
else:  
 print(a, " is divisible by either 2 or 3")  
 print("Hence it satisfies both conditions,", a, "is also divisible by 6")  
# CONDITIONAL STATEMENTS AND LOOPS  
  
# LOAN ELIGIBILITY CHECK (if elif else)  
  
cibil = int(input("Enter cibil: "))  
income = int(input("Enter income: "))  
if cibil >= 750:  
 print("Yayyyy.. you are eligible for a loan. start ruining your life by taking a loan for everything.")  
elif income > 100000:  
 print("Yayyyy.. you are eligible for a loan. start ruining your life by taking a loan for everything.")  
else:  
 print("Our company doesn't give loans to losers.. build your credit score or get a better paying job:)")  
  
# EXECUTING PROGRAM UNTIL CONDITION IS MET (while loop, break statement)  
  
secret\_word = "Among us"  
counter = 0  
  
while True:  
 word = input("Enter the secret word: ")  
 counter = counter + 1  
 if word == secret\_word:  
 print("Welcome Noob!")  
 break  
 if word != secret\_word and counter > 2:  
 print("Hey imposter.. Better Luck Next Time")  
 break  
  
# STRING OPERATIONS  
  
string = "hi welcome to python"  
  
string.upper() # UPPER CASE  
string.lower() # LOWER CASE  
string.capitalize() # CAPITALIZES FIRST LETTER OF THE STRING  
string.title() # CAPITALIZES FIRST LETTER OF EVERY WORD  
string.swapcase() # SWAPS UPPER CASE TO LOWER AND VICE VERSA  
string.split() # RETURNS A LIST CONTAINING ELEMENTS AS WORDS OF THE STRING  
string.find() # SEARCHES FOR SPECIFIC WORD IN A STRING AND RETURNS INDEX  
string.endswith() # TAKES A PARAMETER AND RETURNS A BOOLEAN VALUE IF THE STRING ENDS WITH THE PARAMETER  
string.count() # COUNTS THE RECURRENCE OF THE GIVEN PARAMETER AND RETURNS INTEGER VALUE  
string.replace() # TAKES THREE PARAMETERS AND REPLACES THE SECOND WITH FIRST WITH SECOND AND COUNT  
  
# EXCEPTION HANDLING WITH TRY, EXCEPT AND FINALLY  
  
import sys  
  
name = input()  
ac\_no = input()  
try:  
 if int(ac\_no).isnumeric():  
 print()  
except:  
 print("Sorry!", sys.exc\_info()[0], "occured.")  
finally:  
 print(name, ':', ac\_no)  
  
# WITH RAISING AN EXCEPTION  
  
name = input()  
age = int(input())  
try:  
 if age < 18:  
 raise Exception("not eligible to vote")  
 print("opted for vote")  
except Exception as e:  
 print(e)  
  
# LIST OPERATIONS  
  
list1 = [1,2,3,4,5]  
list2 = ['a','b','c','d']  
# APPEND  
  
list1.append(6) # ADDS THE ELEMENT TO THE END OF THE LIST  
list1.extend([7, 8, 9]) # ADDS MORE THAN ONE ELEMENT TO THE LIST  
list1[3] #INDCATES THE CORRESPONDING ELEMENT IN THE LIST OF THE INDEX  
print(list1[:4]) #SLICING THE LIST ACCORDING TO THE INDEX  
list1.reverse() #REVERSES THE ELEMENTS IN THE LIST  
len(list1) #INDICATES THE LENGTH OF THE LIST  
min(list1) #INDICATES MIN VALUE OF THE LIST  
print (list1+list2) #CONCATENATES BOTH LISTS AND RETURNS A NEW LIST  
list1.sort() #SORTS THE ELEMENTS IN ASCENDING ORDER  
  
#LINKED LIST  
class node:  
 def \_\_init\_\_(self, data):  
 self.data = data  
 self.link = None  
  
 #CREATING LINKED LIST  
class linkedlist:  
 def\_\_init\_\_(self):  
 self.head = None  
  
# TUPLE  
  
tup = (1,2,3,4,5)  
  
tup.count(22) # PRINTS THE FREQUENCY OF OCCURENCE IN THE TUPLE  
tup.index(2) # INDICATES THE POSITION OF THE ELEMENT IN THE TUPLE  
  
#DICTIONARY  
  
dict1 = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25} #CREATING A DICTIONARY (SQUARES)  
for i in dict1:  
 print(dict1[i]) #ITERATING THROUGH A DICTIONARY  
print(dict1.pop(4)) #REMOVES THE PARTICULAR ELEMENT  
  
squares.clear() #REMOVE ALL ELEMENTS  
  
del squares #REMOVE THE WHOLE DICTIONARY  
  
dict1\_descending = {k: d[k] for k in sorted(d, key=d.get, reverse=True)} #SORTING DICTIONARY USING KEYS  
  
#SET  
  
newset = {1, 2, 3, 4} #CREATING SETS  
newset2= {4,5,6,7}  
newset.update(("sai", "Praneeth")) #ADDING AN ELEMENT IN A SET  
  
newset.remove(("Praneeth")) #REMOVING ELEMENT FROM SET  
  
newset.union(newset2) #UNION OF A SET ('|' OPERATOR CAN ALSO BE USED)  
  
newset.intersection(newset2) #INTERSECTION OF A SET ('&' OPERATOR CAN ALSO BE USED)  
  
newset.difference(newset2) # DIFFERENCE OF A SET ('-' OPERATOR CAN BE USED)