# Variables and Operators

Q1: Write a program to find area of square.

```
l = int(input("Enter the length of the square: "))
a = l*l
print("The area of the square is ",a)
```

Q2: Write a program to find area of rectangle.

```
l = int(input("Enter the length of the rectangle: "))
b = int(input("Enter the breadth of the rectangle: "))
a = l*b
print("The area of the rectangle is ",a)
```

Q3: Write a program to find area of triangle.

```
h = int(input("Enter the height of the triangle: "))
b = int(input("Enter the base of the triangle: "))
a = 1/2*(b*h)
print("The area of the triangle is ",a))
```

Q4: Write a program to find perimeter of square.

```
l = int(input("Enter the length of the square: "))
p = 4*l
print("The perimeter of the square is ",p)
```

Q5: Write a program to find perimeter of rectangle.

```
l = int(input("Enter the length of the rectangle: "))
b = int(input("Enter the breadth of the rectangle: "))
p = 2*l+2*b
print("The area of the rectangle is ",p)
```

Q6: Write a program to find perimeter of triangle.

```
a = int(input("Enter first side of triangle: "))
b = int(input("Enter second side of triangle: "))
c = int(input("Enter third side of triangle: "))
p = a+b+c
print("The perimeter of triangle is ",p)
```

Q7: Write a program to find average and percentage.

```
  first_name = input("enter your first name:")
  last_name = input("enter your last name:")
  print("Hello",first_name,last_name,",welcome to average and
percentage calculator application")
```

```
print("Enter marks obtained in 5 subjects: ")
m1 = int(input("Subject 1:"))
m2 = int(input("Subject 2:"))
m3 = int(input("Subject 3:"))
m4 = int(input("Subject 4:"))
m5 = int(input("Subject 5:"))
sum = m1 + m2 + 3 + m4 + m5
average = sum/5
percentage = (sum/500)*100
print("Average Marks = ", average)
print("Percentage Marks = ", percentage, "%\n")
```

# Conditions and Loops

Q1: Write a program to convert decimal to binary , octal and hexadecimal.

```
n = int(input("enter number:"))
print("1 : bin")
print("2 : oct")
print("3 : hex")
choice = int(input("enter your choice:"))
if choice == 1:
    print(format(n,'b'))
elif choice == 2:
    print(format(n,'o'))
elif choice == 3:
    print(format(n,'x'))
else:
    print("invalid choice")
```

Q2: Wriite a program to find Square Root of a Number.

```
  while True:
  print("Enter 'x' for exit.")
  num = input("Enter a number: ")
  if num == 'x':
          break
  else:
          number = float(num)
          number_sqrt = number ** 0.5
          print("Square Root of %0.2f is %0.2f" %(number,
number_sqrt))
```

Q3: Write a program to check number divisible.

```
num = int(input("enter number:"))
x = int(input("entered number is divisible by:"))
if num%x == 0:
   print("YES")
else:
   print("NO")
```

Q4: Write a program to find largest of 3 numbers.

```
while True:
   print("Enter 'x' for exit.")
   print("Enter any three numbers: ")
   num1 = input()
   num2 = input()
   num3 = input()
   if num1 == 'x':
```

```
            break
    else:
            number1 = int(num1)
            number2 = int(num2)
            number3 = int(num3)
            largest = number1
            if largest < number2:
                    if number2 > number3:
                            largest = number2
                    else:
                            largest = number3
            elif largest < number3:
                    if number3 > number2:
                            largest = number3
                    else:
                            largest = number2
            else:
                    largest = number1
            print("Largest of given three numbers
is",largest,"\n")
```

Q5: Write a program to Print ASCII Values in a given range.

```
for i in range(1, 255):
   ch = chr(i)
   print(i, "=", ch)
```

Q6: Write a program to print Fibonacci series.

```
nterms = int(input("how many terms?"))
n1=0
n2=1
count=2
if(nterms<=0):
   print("enter a positive number")
elif(nterms==1):
   print("fibonacci:",n1)
else:
   print("fibonacci:")
   print(n1,n2,  end=" ")
   while count<nterms:
      nth = n1+n2
      print(nth, end=" ")
      n1=n2
      n2=nth
      count += 1
```

Q7: Write a program to display power of numbers.

```
num = int(input("I want to find the power of :"))
```

```
x = int(input("Up to how many terms :"))
for i in range(x):
   print("%i raised to the power %i = %i" %(num,i,num**i))
```

Q8: Write a program to Convert Fahrenheit to Celsius.

```
while True:
   print("Enter 'x' for exit.")
   fah = input("Enter Temperature in Fahrenheit: ")
   if fah == 'x':
           break
   else:
           fahrenheit = float(fah)
           celsius = (fahrenheit-32)/1.8
           print("Temperature in Celsius =",celsius,"\n")
```

# Strings

### Q1: Write a program to count vowels in a country name.

```
name=input("Enter country name:")
name=name.lower()
count=0
for i in name:
  if i in ('a','e','i','o','u'):
    count+=1
print("Number of vowels present in country name:",count)
```

### Q2: Write a program to check whether a string is an anagram.

```
  string1=input("Enter first string:")
  string2=input("Enter second string:")
  string1=string1.lower()
  string2=string2.lower()
  count=0
  for i in range(0,len(string1),1):
    if i in range(0,len(string2),1):
      print("Letter %s in first string is present in second
string"%(string1[i]))
      count+=1
  if count==len(string1):
    print("1st string is anagram of 2nd string")
```

### Q3: Write a program to check whether a string is a palindrome.

```
def ispalin(str):
  if len(str)<=1:
    return "Given string is palindrome"
  else: return str[0]==str[-1]and palin(str[1:-1])

string=input("Enter a string :")
print(ispalin(string))
```

### Q4: Write a program to demonstrate Delete and Replace methhods of strings.

```
#4.1 deleting word from sentence

sentence=input("Enter a sentence:")
sentence=sentence.split()
word = input("Enter word to be delete: ")
print("\nDeleting given word from the given string...")
print("New String after successfully deleted the given word:")
print(' '.join([i for i in sentence if i not in word]))
```

```
  print("\n")
```

```
#4.2 replacing string in given sentence

  sentence=input("Enter a sentence:")
  word=input("Enter the word which you wanna replace:")
  new_word=input("Enter a word to replace:")
  if word in sentence:
    sentence=sentence.replace(word,new_word)
  else:print("Word not found")
  print("New sentence is",sentence)
```

Q5: Write a program to demontrate dir, stratswith and endswith methods of strings.

```
  #5.1 dir()

  string="xyz"
  for i in dir(string):
    print(i)

  #5.2 dir() startswith() ignoring the methods that startswith
"__"

  string="xyz"
  for i in dir(string):
    if not i.startswith("__"):
      print(i)

  #5.3 dir() endswith() ignoring the methods that endswith "__"

 string="xyz"
 for i in dir(string):
    if not i.endswith("__"):
      print(i)
```

Q6: Write a program to validate IP address using split method.

```
  def ip_validator(ip):
  ip=ip.split(".")
  if len(ip)!=4:
    return "Invalid IP"
  elif len(ip)==4:
    for i in ip:
      ip=".".join(ip)
      if not int(i) in range(0,256):
        return "%s is Inalid IP" %(ip)
      else:return "%s is Valid IP" %(ip)
  ip=input("Enter ip address:")
  print(ip_validator(ip))
```

Q7: Write a program to correct time in HH:MM format using zfill method.

```
time=input("Enter time in hour:")
time=time.split(":")
time_hr=time[0]
time_hr=time_hr.zfill(2)
time2=time[1].split()
time_am_pm=time2[1]
time_min=time2[0]
time_min=time_min.zfill(2)
new_time=time_hr+":"+time_min+" "+time_am_pm
print(new_time)
```

Q8: Write a program to encrypte a giving string using your own key.

```
d={"a":"b","b":"c","c":"d","d":"e","e":"f","f":"g","g":"h","h":"i
","i":"j","j":"k","k":"l","l":"m","m":"n","n":"o","o":"p","p":"q"
,"q":"r","r":"s","s":"t","t":"u"}
 string="abc"
 l=[]
 for i in string:
   l.append(d.get(i))
 new_string="".join(l)
 print(new_string)
```

Q9: Write a program to validate a password whose minimum length is 7 and contains atleast one digit and one special character.

```
 "PASSWORD MUST CONTAIN ONE DIGIT AND ONE SPECIAL CHARACTER !!!"
 password=print("Enter password to validate:")
 if len(password)>=7:
   for i in password :
     if  i in ("!","@","#","$","%","^","&","*","(",")" and
i.isalnum()==True):
       flag = 1
     else:
       flag=0
 else:
   flag = 0
 if flag==1:
   print("Valid password")
 elif flag==0:
   print("Invalid password")
```

Q10: Write a program to check the frequency(number of times repeated) of word 'because' in a given string.

```
speech=input("Enter your speech:")
speech=speech.lower()
speech=speech.split()
count=0
for i in speech:
  if i=="because":
    count+=1
```

```
print("Number of 'because' present in speech are:",count)
```

# List and Tuple

Q1: Write a program to count the number of odd numbers in a list.

```
lst = [1,2,3,4,5,6]
count = 0
for i in lst:
    if i%2!=0:
        count += 1
print(count)
```

Q2: Write a program to add the number of odd numbers in a list.

```
lst = [1,2,3,4,5,6]
sum1 = 0
for i in lst:
    if i%2!=0:
        sum1 += i
print(sum1)
```

Q3: Write a program to select odd numbers in a list.

```
lst = [1,2,3,4,5,6]
lst1 = []
for i in lst:
    if i%2!=0:
        lst1.append(i)
print(lst1)
```

Q4: Write a program to select comman elements in a given two list.

```
lst1 = [1,2,3,4,5]
lst2 = [1,4,7,9]
lst3 = []
for i in lst1:
    if i in lst2:
        lst3.append(i)
print(lst3)
```

Q5: Write a program to combine elements of two lists into one list.

```
lst1 = [1,2,3,4,5]
lst2 = [1,4,7,9]
lst3 = lst1+lst2
print(lst3)
```

Q6: Write a program to demonstrate difference between two lists.

```
lst1 = [1,2,3,4,5]
```

```
lst2 = [1,4,7,9]
lst3 = []
for i in lst1:
    if i not in lst2:
        lst3.append(i)
print(lst3)
```

Q7: Write a program to delete first and last element in a list.

```
lst = [1,2,3,4,5]
lst.pop(0)
lst.pop(-1)
print(lst)
```

Q8: Write a program to demonstrate list of list.

```
lst=[[1,2,3,4],[78,96,86],[10,20,30]]
print(lst)
#to print first element of the first list
print(lst[0][0])
#to print second element of the third list
print(lst[2][1])
```

Q9: Write a program to deonstrate list comprhension.

```
v = [2**i for i in range(13)]
print(v)
s = [x for x in range(10)]
m = [x for x in s if x % 2 == 0]
print(s)
print(m)
```

Q10: Write a program solve last man stanidng puzzle.

```
sol = int(input("Enter the number of soldiers :"))
list1=[]
for item in range(1,sol+1):
    list1.append(item)
i=0
while(len(list1)!=1):
    list1.pop(i+1)
    i+=1
    if(i==len(list1)):
        i=0
    if(i==len(list1)-1):
        i=-1
print(list1[0])
```

# Dictionary

Q1: Write a program to sort keys in a dictionary.

```
dic = {1:'a',3:'b',2:'c',20:'d',10:'e'}
lst = list(dic.keys())
lst.sort()
print(lst)
```

Q2: Write a program to demonstrate dictionary comprehension.

```
dic = {x: x**3 for x in range(10)}
print(dic)
```

Q3: Write a program to map keys with element.

```
dic = {1:'a',3:'b',2:'c',20:'d',10:'e'}
print(dic)
k = int(input("Enter any one key from the above dictionary :"))
a = dic[k]
print("The value associated with the entered key is: ",a)
```

Q4: Write a program to demonstrate the use of dic.items method.

```
dic = {1:'a',3:'b',2:'c',20:'d',10:'e'}
for items in dic.items():
    print(items)
```

Q5: In gene expression, mRNA is transcribed from a DNA template. The 4 nucleotide bases of A, T, C, G corresponds to the U, A, G, C bases of the mRNA. Write a function that returns the mRNA transcript given the sequence of a DNA strand.

```
def mRNAtranscription(dna_template):
  dna2rna = {"A":"U","T":"A","C":"G","G":"C" }
  mRNA = ''
  for base in dna_template:
    mRNA=mRNA+dna2rna.get(base)
  return mRNA
```

Q6: Write a function invertDictionary(d) that takes in a dictionary as argument and return a dictionary that inverts the keys and the values of the original dictionary.

```
def invertDictionary(d):
  d1={}
  for k,v in d.items():
    l=[]
    if (d1.get(v)):
      for values in d1.get(v):
        l.append(values)
    l.append(k)
    d1[v]=l
  return d1
```

Q7: Write a function countLetters(word) that takes in a word as argument and returns a dictionary that counts the number of times each letter appears.

```
def countLetters(word):
  return {i:word.count(i) for i in word}
```

Q8: A DNA strand consisting of the 4 nucleotide bases is usually represented with a string of letters: A,T, C, G. Write a function that computes the base composition of a given DNA sequence.

```
def baseComposition(dna_seq):
  return {j:dna_seq.count(j) for j in ('A','C','T','G') }
```

Q9: Write a function reverseLookup(dictionary, value) that takes in a dictionary and a value as arguments and returns a sorted list of all keys that contains the value. The function will return an empty list if no match is found.

```
def reverseLookup(dictionary, value):
  list=[]
  for i in dictionary.keys():
    if (dictionary.get(i)==value) :
      list=list+[i]
  return sorted(list)
```

Q10: A sparse vector is a vector whose entries are almost all zero, like [1, 0, 0, 0, 0, 0, 0, 2, 0]. Storing all those zeros wastes memory and dictionaries are commonly used to keep track of just the nonzero entries. For example, the vector shown earlier can be represented as {0:1, 7:2}, since the vector it is meant to represent has the value 1 at index 0 and the value 2 at index 7. Write a function that converts a sparse vector into a dictionary as described above.

```
def convertVector(numbers):
  d={}
  for i in range (0,len(numbers)):
    if numbers[i] == 0:
      continue
    else:
      d[i]=numbers[i]
  return d
```