

Computer Science

Class XI- 2021

[Question Bank for Easy Revision – Part 1]

Important functions from packages/modules:

(i) **Math (module):**

ceil(), exp(), factorial(), floor(), gcd(), log(), log10(), log2(), pow(), remainder(), sqrt(), trunc(), sin(), cos(), tan(), radian(), degrees(), pi()

(ii) **random (module):**

randint(), random(), randrange(), uniform()

(iii) **statistics (module):**

mean(), median(), mode()

(iv) **csv (module):**

reader(), writer()

(v) **pickle (module):**

dump(), load()

(vi) **mysql.connector / MySQLdb.connector**

(connector package from mysql / MySQLdb package)

connect()

cursor() and commit() are methods belong to connection object which is created by the connect() method from connector package.

(vii) **matplotlib.pyplot** **Not in syllabus for 2021 Exams** **(pyplot module from matplotlib package)**

Axis(), bar(), barh(), plot(), pie(), show(), title(), xlabel(), ylabel()

Descriptive Questions:

- What is `print()` in Python?
- Name any two keyword arguments of `print()` function.
- What is a data type? What are the fundamental data types in Python?
- Identify the data types of the following data objects:
(i) 7 (ii) '7' (iii) "7" (iv) 7.0 (v) '''7''' (vi) -0.7 (vii) -29 (viii) -15.38
(ix) "29 acres" (x) ". & #"
 - Find the data type of number 12.
 - Find the data type of string 'Hello'
 - Find the data type of value represented by a variable `Hello`.
 - Find the address of memory location represented by a variable `amount`.
 - Display the string `'-'` 25 times in a line.
- What are unary operators and binary operators? Give two examples of each.
- What is the operator precedence of arithmetic operators in Python?
- Evaluate the following expressions:
(i) $(2 + 3) ** 3 - 6 / 2$ (ii) $(2 + 3) * 5 // 4 + (4 + 6) / 2$ (iii) $12 + (3 * 4 - 6) / 3$
(iv) $12 + (3 * 4 - 6) // 2$ (v) $12 * 3 \% 5 + 2 * 6 // 4$ (vi) $12 \% 5 * 3 + (2 * 6) // 4$
- What is a variable? What are the rules for forming valid variable names in Python?
- Which function is used to know the address of the memory location to which a variable is referring?
- Write Python statements to display a list of Python keywords.
- Why is Python called a dynamic type language?
- What is an augmented assignment operator? Give an example to explain.
- Identify invalid variable names from the following, give reason for each:
`Group, if, int, total marks, S.I., volume, tot_strength, 9t, tag$`
- Write Python expressions equivalent to the following arithmetic/algebraic expressions:
(a) $\frac{\square + \square}{2}$ (b) $\frac{3^2 + 9^3}{2}$ (c) $3^2 + \frac{9^3}{5}$
(d) $\sqrt{\square} + \frac{\square + 2}{\square}$ (e) $8 - 6 + \frac{6 * \square \square \square}{7} - \sqrt{\square \square \square}$ (f) $\square \square + \frac{1}{2} \square \square^2$
(u, a, t are variables)
- Write Python expressions to:
(a) represent the Integral part of the quotient when 63 is divided by 29.
(b) represent the Remainder when 63 is divided by 29.
(c) represent the Fourth root of 3
(d) Add the remainder of 6/5 to the product of 6 and 5.
(e) represent Square root of the sum of 8 and 43.
(f) represent Sum of the square roots of 8 and 4³.
- What are comments in a program? How are comments given in a Python script? Give an example.
- Define Syntax error, Run time error, and Logical error. Also give a suitable example of each.
- What is a token? What are different categories of tokens in Python? Give two examples of tokens of each category.
- Observe the following script and enlist all the tokens used in it. For each token, write its category too:

```
#Identify tokens in the script
name="Arjun"
age=eval(input("Enter age: "))
print(name,"is",age,"years old")
```
- Evaluate the following logical conditions assuming that `a=10, b=20, c='three', d='four'`
(i) `a>b/2` (ii) `a<=b/2` (iii) `a!=b` (iv) `b==a**2` (v) `a+b > -b*-a`
(vi) `b*2 > ord("2")` (vii) `c==d` (viii) `c>d` (ix) `c<d` (x) `"c"<"d"`
(xi) `"c"==c` (xii) `d != chr(ord("d"))`

22. What are relational operators? What relational operators are available in Python?
23. What are logical operators? What logical operators are available in Python?
24. Write Python expression for each of the following conditions (assuming that num1, num2 are numbers):
 - (i) num1 is equal num2.
 - (ii) num1 is unequal to num2.
 - (iii) num1 is a factor of num2.
 - (iv) num1 is a multiple of num2.
 - (v) both num1 and num2 are even.
 - (vi) both num1 and num2 are odd.
 - (vii) one of these numbers (num1 and num2) is even and the other is odd.
 - (viii) the sum of num1 and num2 is greater than the product of num1 and num2.
 - (ix) num1 is a multiple of either 3 or 5 or both
 - (x) num1 is a multiple of 3 and 5.
25. When does the else clause of a loop get executed? Is else clause mandatory in a loop?
26. What is the use of break and continue statements in a loop? Give a suitable example to explain.
27. What is an infinite loop? Give an example of such a loop.
28. Write Python statements to create the following lists:
 - (i) A list of 5 fruit names and name it as fruit.
 - (ii) A list of 5 city names and name it as City.
 - (iii) A list containing 3 strings and 2 integers and name it as Mixed.
 - (iv) A list containing 3 integers and a list of 2 strings and name it as Box.
 - (v) A list of 10 random integers each in the range 10 to 50 and name it as R10.
 - (vi) A list of 10 random floating point numbers in the range 10.00 to 49.99 and name it as F10.
 - (vii) A list of 10 values of the expression x^2+2x-1 , for integer values of x from -4 to 5 and name it as y.
 - (viii) A list of 10 random integers in the range -30 to -20 and name it as C.
 - (ix) A list, named A, containing 3 numbers, and another list, named B, containing list A and 3 numbers.
 - (x) A list, named A, containing 3 numbers, and another list, named B, containing list A and another unnamed list of 3 integers.
29. Write a Python statement to:
 - (i) Replace the last element of a list L1 with a list L2
 - (ii) Replace the last element of a list L1 with the elements of a list L2.
30. Consider the lists A and B given below and state the values (True/False) of the expressions that follow:


```
A=[5, 3, 8]
B=[A, 8, 66, 45]
```

 - (i) `A in B` (ii) `'A' in B` (iii) `[A] in B` (iv) `[5, 3, 8] in B`
 - (v) `8 in B` (vi) `[8] in B` (vii) `[A, 8] in B` (viii) `66 in B`
 - (ix) `45 in B` (x) `A not in B`
31. Differentiate between mutable and immutable data objects. Give an example of each.
32. Which methods in Python are available for both lists and tuples?
33. Write Python statements to create the following tuples:
 - (i) A tuple of 5 fruit names and name it as fruit.
 - (ii) A tuple of 5 city names and name it as City.
 - (iii) A tuple containing 3 strings and 2 integers and name it as Mixed.
 - (iv) A tuple containing 3 integers and a list of 2 strings and name it as Box.
 - (v) A tuple of 10 random integers each in the range 10 to 50 and name it as R10.
 - (vi) A tuple of 10 random floating-point numbers in the range 10.00 to 49.99 and name it as F10.
 - (vii) A tuple of 10 values of the expression x^2+2x-1 , for integer values of x from -4 to 5 and name it as y.
 - (viii) A tuple of 10 random integers in the range -30 to -20 and name it as C.

- (ix) A list, named A, containing 3 numbers, and a tuple, named B, containing list A and 3 numbers.
- (x) A tuple, named A, containing 3 numbers, and another tuple, named B, containing tuple A and another unnamed tuple of 3 integers.
34. Consider the list A and tuple B given below and state the values (True/False) of the expressions that follow:
`A=[5, 3, 8], B=(A, 8, 66, 45)`
- (i) `A in B` (ii) `'A' in B` (iii) `[A] in B` (iv) `[5,3,8] in B`
 (v) `8 in B` (vi) `[8] in B` (vii) `[A, 8] in B` (viii) `66 in B`
 (ix) `45 in B` (x) `A not in B`
35. Write Python statements to create the following dictionaries:
- (i) A dictionary with 5 fruit names as keys and their respective prices as values.
- (ii) A dictionary with 3 student names as keys and their aggregate marks percentages as respective values.
- (iii) A dictionary named student with the following elements:
 a. **admno**: admission number of the student (string)
 b. **name**: name of the student (string)
 c. **dob**: Date of birth of the student (string)
 d. **class**: Class of the student (integer)
- (iv) A dictionary named item with the following elements:
 a. **code**: item code (integer)
 b. **name**: name of the item (string)
 c. **price**: price of the item (floating point)
 d. **qty**: balance quantity of the item in stock (floating point)
- (v) A dictionary named mobile with the following elements:
 a. **make**: Name of the brand (Apple, Samsung, etc.)
 b. **model**: Model name (iPhone X, Note 9, etc.)
 c. **year**: Year of launch (integer)
 d. **price**: Price in INR (integer)
- (vi) A dictionary Eng2Lang with 5 elements, where key is an English language word and value is the corresponding word in your mother tongue.
- (vii) A dictionary to store the vertices of a polygon. Each element of the dictionary should have a vertex name as the key and a tuple to store the coordinates of the vertex as its value. For example, an element of the dictionary may be 'A':(2,3)
36. Consider dictionary D given below:
`D={'U1':'Programming', 'U2':'Data structures', 'U3':'Files', 'U4':'RDBMS', 'U5':'Networks'}`
- With reference to D, state True/False for each of the following expressions:
- (i) `'U2' in D` (ii) `U2 in D` (iii) `'Programming' in D`
 (iv) `Programming in 'D'` (v) `'U3' not in D` (vi) `'Files' not in D`
37. What is a string? How many types of strings are supported in Python? Give an example of each.
38. Consider the strings A and list B given below and state the values (True/False) of the expressions that follow:
`A='Application'`
`B='An Apple A Day'`
- (i) `A in B` (ii) `'A' in B` (iii) `[A] in [B]` (iv) `A[:3] in B`
 (v) `A[3] in B` (vi) `A[3:] in B` (vii) `B in A` (viii) `A not in B`

Correcting the code:

- Find error(s), if any, in the following statements:
 (i) `print("three")` (ii) `Print(3)` (iii) `print(44+'56')` (iv) `print(3,2 sep=':')`
 (v) `print "wisdom"` (vi) `print['33+44']` (vii) `PRINT("15 August 1947")`
- Find error(s), if any, in each of the following function calls:
 (i) `input(Enter a number: )` (ii) `input("Enter your age: ")` (iii) `int(56)`

(iv) <code>int("56.6")</code>	(v) <code>int("fifty6")</code>	(vi) <code>int('5six')</code>	(vii) <code>int(5six)</code>
(viii) <code>float(12.3)</code>	(ix) <code>float("-12.45")</code>	(x) <code>float(5)</code>	(xi) <code>int(5/4)</code>
(xii) <code>float("5/4")</code>	(xiii) <code>eval(10)</code>	(xiv) <code>eval('10')</code>	(xv) <code>eval('10+2')</code>
(xvi) <code>eval(10*2)</code>	(xvii) <code>int(eval("10/3"))</code>	(xviii) <code>eval(int(10/3))</code>	
(xix) <code>eval(int("10/3"))</code>		(xx) <code>eval(float("10/3"))</code>	

3. Find error(s), if any, in each of the following scripts:

```
(i)      n1,n2=5,4
        if n1=n2:
            print("Equal")

(ii)     n1=n2=12/5
        if n1!=n2
        print"Not equal"
        else print"Equal"

(iii)    12=age
        if age<=10:
            print(Primary)
        elseif age<=13:
            print(Middle)
        elseif age<=15:
            print(Secondary)
        else age<=17:
            print(Sr Secondary)

(iv)     num=input("Enter an integer: ")
        if num<0:
            num=-num;
        if num>=0 AND num<=9
            print("Single digit number")
        elif: num>=10 OR num<=99
            print("Two digit number")
        elif: num<=999
            print("Three digit number")
        else: print("Number contains more than 3 digits")

(v)      a,b,c=12,13,14
        if (a>=b or a<=c and a+c<=b):
        print("A")
        ELSE print(a+b+c)
```

4. Each of the following code segments is written to perform the job specified in the respective comment. Check each of these code segments and make the necessary changes to correct the code:

```
(i)      #To display "Hello" 10 times
        for i in range(1, 10):
            print("Hello")

(ii)     #To print first 10 natural numbers
        i=1
        while (i==10):
            print("i")
            i+=1

(iii)    #To display first 10 even natural numbers
        num=1
        while (num<=10):
            print(num)
            num+=1

(iv)     #To display first 10 odd natural numbers
```

```

        for i in range (10):
            print(i)
(v) #To calculate DA as 10% of basic salary for 10 employees
    basic=input("Enter basic salary")
    da=10*basic
    print("DA =",da)

```

5. Find syntax error(s), if any, in each of the following code segments:

```

(i) a=int(input("Enter a number: "))
    c==1
    for b in range (a,a*10.0,a)
        c=*b
        print(a,b,c)
(ii) a=int(input("Enter a number: "))
    b=int(input("Enter another number: "))
    start,end=a,b
    if a>b:
        start,end=b,a
    i=start
    while i<=end
        if (i%start==0):
            print(i,end='--',start)
            i+=1
    print(i)
(iii) a=int(input("Enter a number: "))
    b=int(input("Enter another number: "))
    if a>b: step=-1
    else: step=1
    while a!=b:
        if (a+b%2>0):
            print(a+b,sep="--")
            a+=step
(iv) a=int(input("Enter a number: "))
    while a%5!=0
        if (a%2>0):
            print(a,sep="--")
            a+=1
    else: print(a)
(v) x=123045
    while x%10:
        x//=10
        print(x,sep="--")
        else print(a)
(vi) x=123045
    while x%10:
        d=x%10
        for i in range(d):
            print(i)
        else: print (d)
        print(x,sep='--')
        x/=10
    else print(a)

```

6. Identify and correct the errors, if any, in the following code segments, and underline each correction.

```

(i) a=[1,2,3,2,1]
    for x in range(a):
        print(x)

(ii) a=["Umang","Usman","Utkarsh","Umedh"]

```

- ```

 for x in range(len(a)):
 print(a(x))

(iii) A=5*(None)
 A=8*[3]
 for i in range(5):
 A[i]=eval(A[i]*i)
 for i in A:
 print(A(i))

```
7. Find error(s), if any, in the following code segments:
- (i) `a=(1,2,3,2,1)`  
`for x in range(a):`  
`print(x)`
- (ii) `a=("Umang","Usman","Utkarsh","Umedh")`  
`for x in range(len(a)):`  
`print(a(x))`
8. Find error(s), if any, in the following code segments:
- (i) `A="Rajasthan"`  
`A[3]='a'`  
`for i in A:`  
`print(A[i])`
- (ii) `s="Story teller"`  
`if (isupper(s[0]))`  
`print("s starts with upper case letter")`

## Finding the Output:

1. Find outputs of the following commands in Python:
- (i) `print("Hello")`    (ii) `print(2+3, 34-67)`    (iii) `print("2+3", 2+3)`  
 (iv) `print('2+3', 2+3, sep="=")`    (v) `print(2**3, 3**2, sep='*')`  
 (vi) `print(2**3, 2**-3, -2**3, (-2)**3, sep=" and ")`
2. Find the output of each of the following code segments:
- |                                                                                                                                                                                 |                                                                                                                                                                                                 |                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(a) <code>x=3</code><br/> <code>y=x+2</code><br/> <code>x+=y</code><br/> <code>print(x,y)</code></p>                                                                         | <p>(b) <code>x=-2</code><br/> <code>y=2</code><br/> <code>x+=y</code><br/> <code>y-=x</code><br/> <code>print(x,y)</code></p>                                                                   | <p>(c) <code>a=5</code><br/> <code>b=2*a</code><br/> <code>a+=a+b</code><br/> <code>b*=a+b</code><br/> <code>print(a,b)</code></p>                                                            |
| <p>(d) <code>p=10</code><br/> <code>q=20</code><br/> <code>p*=q/3</code><br/> <code>q+=p+q*2</code><br/> <code>print(round(p,2))</code><br/> <code>print(round(q,2))</code></p> | <p>(e) <code>p=5%2</code><br/> <code>q=p**4</code><br/> <code>r=p//q</code><br/> <code>p+=p+q+r</code><br/> <code>r+=p+q+r</code><br/> <code>q-=p+q*r</code><br/> <code>print(p,q,r)</code></p> | <p>(f) <code>p=21//5</code><br/> <code>q=p%4</code><br/> <code>r=p*q</code><br/> <code>p+=p+q-r</code><br/> <code>r*=p-q+r</code><br/> <code>q+=p+q</code><br/> <code>print(p,q,r)</code></p> |
3. Find output of each of the following statements:
- (i) `print(int(56))`    (ii) `print(int("56"))`    (iii) `print(int(5/4))`  
 (iv) `print(float(12.3))`    (v) `print(float("-12.45"))`    (vi) `print(float(5))`  
 (vii) `print(eval('10'))`    (viii) `print(eval('10+2'))`    (ix) `print(int(eval("10/3")))`
4. Find output generated by each of the following scripts:
- (i) `n=1980`



```

if n%100!=0 and n%4==0 or n%400==0:
 print("Leap year")
else: print("Not a Leap year")

```

(ii) n1=n2=12/5

```

if int(n1)==n2:
 print("Integer")
else: print("Non integer")

```

(iii) age=12

```

if not age<18:
 print("Eligible to vote")
else: print("Not eligible to vote")

```

(iv) num=int(input("Enter an integer: "))  
 #find output for num=-245, 4, 56, and 564  
 if num<0:  
 num=-num;  
 if num>=0 and num<=9:  
 print("Single digit number")  
 elif num>=10 or num<=99:  
 print("Two digit number")  
 elif num<=999:  
 print("Three digit number")  
 else: print("Number contains more than 3 digits")

(v) a,b,c=12,13,14

```

if (a>=b or a<=c and a+c<=b):
 print("A")
else: print("B")

```

5. Find output generated by each of the following scripts:

(i) a=4

```

c=0
for b in range (a,a*a,a):
 c+=b
 print(a,b,c)

```

(ii) #Find output for (i) a,b=4,10 (ii) a,b=2,-4

```

a=int(input("Enter a number: "))
b=int(input("Enter another number: "))
start,end=a,b
if a>b:
 start,end=b,a
i=start
while i<=end:
 if (i%start==0):
 print(i,end='--')
 i+=1
print(i)

```

(iii)

```

a=10
b=a-6
if a>b:
 step=-1
else: step=1
while a!=b:
 if (a+b%2>0):
 print(a+b,sep="--")
 a+=step

```

(iv) #Find output for (i) a=10 (ii) a=-6

```

a=int(input("Enter a number: "))

```

```

while a%5!=0:
 if (a%2>0):
 print(a, sep="--")
 a+=1
 else: print(a)
(v) x=123045
while x%10:
 x//=10
 print(x, sep="--")
else: print(x)
(vi) x=123045
while x%10:
 d=x%10
 for i in range(d):
 print(i)
 else: print (d)
 print(x, sep='--')
 x//=10

```

6. Find output generated by each of the following scripts:

```

(i) for i in range(4):
 for j in range(i,1,-1):
 print(i,j)

(ii) for i in range(4):
 for j in range(i,1,-1):
 print(i,j,end='*')
 print(i,j)
print(i,j)

(iii) print("-"*10)
for i in range(3):
 if i%2:
 for j in range(i,1,-1):
 print(i,j)
 else: print(i,j)
 else: print(i)
else: print(i,j)

```

7. Consider list A given below and write the output of the statements that follow:

A=[3, 6, 7, 8, 10, 12, 23, 33, 16, 6, 2, 1]

|                          |                          |                           |
|--------------------------|--------------------------|---------------------------|
| (i) print(len(A))        | (ii) print(A[3])         | (iii) print(A[-3])        |
| (iv) print(A[3:])        | (v) print(A[-3:])        | (vi) print(A[::3])        |
| (vii) print(A[3::])      | (viii) print(A[3::-2])   | (ix) print(A[-3::-2])     |
| (x) print(A[::-3])       | (xi) print(A[3:-3:3])    | (xii) print(A[3:-3:-3])   |
| (xiii) print(len(A[3:])) | (xiv) print(len(A[-3:])) | (xv) print(len(A[:::-3])) |

8. Write output of the following code segment:

```

a=[1,2,3,2,1]
for x in a:
 print(a[x],end='*')

```

9. Write output of the following code segment:

```

A=[x for x in range(2,5)]
B=[-x for x in range(2,5)]
for i in range(len(A)):
 print(A[i]+B[i])
A.extend(B)
B.insert(2,3)
for i in B:

```

```

 if i in A:
 A.remove(i)
print(A, B)
x=A.pop()
y=B.pop(0)
print(x, y)
print(A,B)

```

10. Write output of the following code segment:

```

nums=[1,2,7,9,5,7,2,4,5,1,1,2]
for i in range(1,10):
 if i in nums:
 index=nums.index(i)
 c=nums.count(i)
 print(i,'appears in nums',c,'times.',end=' ')
 print('Its first occurence is at index',index)

```

11. Consider the lists A and B given below and write the output of the statements that follow:

```

A=[5, 3, 8]
B=[A, 8, 66, 45, 'A', ['A',A], ["nested", "lists"]]

```

|                      |                                       |                                    |
|----------------------|---------------------------------------|------------------------------------|
| (i) print(A)         | (ii) print(B)                         | (iii) print(A[0])                  |
| (iv) print(A[0:])    | (v) print(A[:0])                      | (vi) print(B[0])                   |
| (vii) print(B[0][1]) | (viii) print(B[0][1:])                | (ix) print(B[4])                   |
| (x) print(B[5])      | (xi) for i in B:<br>print(I, end='*') | (xii) for i in A:<br>print(B[i%6]) |

12. Write output of the following code segment:

```

import copy
values=[5,3,9,7,1,2,4]
values1=values.copy()
values2=copy.copy(values)
values3=copy.deepcopy(values)
values.sort()
print(values)
print(values1)
print(values2)
print(values3)

```

13. Write output of the following code segment:

```

import copy
a=[7,1,2]
A=[5,3,9,a,4]
B=A.copy()
C=copy.copy(A)
D=copy.deepcopy(A)
a.sort()
print(a)
print(A)
print(B)
print(C)
print(D)
B[3][1]=10
D[3][1]=55
print(a)
print(A)
print(B)
print(C)

```

```
print(D)
```

14. Consider tuple A given below and write the output of the statements that follow:

```
A=(3,6,7,8,10,12,23,33,16,6,2,1)
```

|                          |                          |                          |
|--------------------------|--------------------------|--------------------------|
| (i) print(len(A))        | (ii) print(A[3])         | (iii) print(A[-3])       |
| (iv) print(A[3:])        | (v) print(A[-3:])        | (vi) print(A[:3])        |
| (vii) print(A[3::-1])    | (viii) print(A[3::-2])   | (ix) print(A[-3::-2])    |
| (x) print(A[::-3])       | (xi) print(A[3:-3:3])    | (xii) print(A[3:-3:-3])  |
| (xiii) print(len(A[3:])) | (xiv) print(len(A[-3:])) | (xv) print(len(A[::-3])) |

15. Write output of the following code segment:

```
a=(1,2,3,2,1)
for x in a:
 print(a[x], end='#')
```

16. Write output of the following code segment:

```
A=(2,3,4,5,6)
B=-2,-1,0,1,2
for i in range(len(A)):
 print(A[i]+B[i])
B=B[:2]+(3,)+B[2:]
for i in B:
 if i in A:
 print(i)
print(A, B)
x=A[1]
y=B[-1]
print(x, y)
print(A,B)
```

17. Write output of the following code segment:

```
nums=(1,2,7,9,5,7,2,4,5,1,1,2)
for i in range(1,10):
 if i in nums:
 index=nums.index(i)
 c=nums.count(i)
 print(i,'appears in nums',c,'times.',end=' ')
 print('Its first occurrence is at index',index)
```

18. Consider the tuples A and B given below and write the output of the statements that follow:

```
A=(5, 3, 8)
B=(A, 8, 66, 45, 'A',['A',A], ["nested", "tuples"])
```

|                      |                        |                   |
|----------------------|------------------------|-------------------|
| (i) print(A)         | (ii) print(B)          | (iii) print(A[0]) |
| (iv) print(A[0:])    | (v) print(A[:0])       | (vi) print(B[0])  |
| (vii) print(B[0][1]) | (viii) print(B[0][1:]) | (ix) print(B[4])  |
| (x) print(B[5])      | (xi) for i in B:       | (xii) for i in A: |
|                      | print(i)               | print(B[i%6])     |

19. Write output of the following code segment:

```
A=[1,2,3]
B=A
A+=[4]
print(A)
print(B)
A=(1,2,3)
B=A
A+=(4,)
print(A)
print(B)
```

20. Write output of the following code segment:

```
A=[1,2,3]
B=A
A=A+[4]
print(A)
print(B)
A=(1,2,3)
B=A
A=A+(4,)
print(A); print(B)
```

21. Find the output of the following Python code:

```
D={'U1':'Programming', 'U2':'Data structures',
 'U3':'Files', 'U4':'RDBMS', 'U5':'Networks'}
print(len(D))
print(D.pop('U1'))
print(D.popitem())
print(D.pop('U3'))
print(D.get('U4'))
print('U2' in D)
print('Programming' in D)
print('U3' not in D)
print('Files' not in D)
```

22. Consider dictionary weight given below and write the output of the code segments that follow:

```
weight={'A1':8, 'A2':7, 'B1':6, 'B2':5, 'C1':4, 'C2':3, 'D1':2, 'D2':1, 'E':0}
```

- (i) for k in weight:  
    print(k, "-", weight[k])
- (ii) for k in weight.keys():  
    print(k, "-", weight[k])
- (iii) for k in weight.values():  
    print(k, end=' ')
- (iv) for k,v in weight.items():  
    print(k, "-", v)

23. Consider a string A given below and write the output of the statements that follow:

```
A='Passionate Programmer'
```

- |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|
| (i) print(len(A))        | (ii) print(A[3])         | (iii) print(A[-3])       |
| (iv) print(A[3:])        | (v) print(A[-3:])        | (vi) print(A[:3])        |
| (vii) print(A[3::])      | (viii) print(A[3::-2])   | (ix) print(A[-3::-2])    |
| (x) print(A[::-3])       | (xi) print(A[3:-3:3])    | (xii) print(A[3:-3:-3])  |
| (xiii) print(len(A[3:])) | (xiv) print(len(A[-3:])) | (xv) print(len(A[::-3])) |

24. Write output of the following code segments:

- (i) a='aaaaa'  
    print(a.count('a'))  
    print(a.count('aa'))  
    print(a.count('aa',2))  
    print(a.count('aa',3))
- (ii) a='mississippi'  
    print(a.count('ss'))

```

 print(a.count('ss',2))
 print(a.count('ss',3))
(iii) a=[1,2,3,4,5]
 b='12345'
 for x in a:
 print(b[x:],a[x:])

(iv) s="United Nations"
 for i in range(len(s)):
 if i%2==0:
 print(s[i],end='')
 elif s[i]>='a' and s[i]<='z':
 print('*',end='')
 elif s[i]>='A' and s[i]<='Z':
 print(s[i:],end='')

(v) P='Dynamic Act'
 Q=''
 Length = len(P)
 for Pos in range(0,Length-2,2):
 Q+=P[Pos+2]
 print(Q)

(vi) Name = "ComPUteR"
 for x in range(0,len(Name)):
 if Name[x].islower():
 print(Name[x].capitalize(), end='')
 elif Name[x].isupper():
 if x%2==0:
 print (chr(ord(Name[x])+32),end='*')
 else:
 print (Name[x-1])

(vii) s="ComPutEr"
 for i in s:
 if i in 'aeiou':
 print('*',end='')
 elif i.isupper():
 print(i,end='')
 else: print('-')

```

25. Write output of the following code segments:

```

(i) s1="Computer"
 s2="Their Computers"
 s3="All 25 Computers In Our Computer Lab"
 i=s2.index(s1)
 j=s3.index(s1)
 k=s3.index(s1,i)
 count=s3.count(s1)
 print(i,j,k,count)

 if s1.isalpha():
 print('s1 is an alphabetic string')
 else: print('s1 is not an alphabetic string')

 if s1.isnumeric():
 print('s1 is a numeric string')
 else: print('s1 is not a numeric string')

 if s1.isalnum():
 print('s1 is an alphanumeric string')
 else: print('s1 is not an alphanumeric string')

```

```

 if s3.isalpha():
 print('s3 is an alphabetic string')
 else: print('s3 is not an alphabetic string')

 if s3.isnumeric():
 print('s3 is a numeric string')
 else: print('s3 is not a numeric string')

 if s3.isalnum():
 print('s3 is an alphanumeric string')
 else: print('s3 is not an alphanumeric string')

 if s3.istitle():
 print('s3 is in Title case')
 else: print('s3 is not in Title case')

(ii) A=', '.join('Split')
 print(type(A), len(A), A)
 B=A.split(',')
 print(type(B), len(B), B)
 A="One, Two, Three"
 B=A.split()
 print(type(B), len(B), B)

(iii) sub="Eng,Eco,CompSc"
 sub=sub.replace("Eco","BSt")
 print(sub)
 sub.swapcase()
 print(sub)

(iv) sub="Eng,Eco,CompSc"
 i=sub.index('o')
 j=sub.find('o')
 k=sub.rfind('o')
 print(i,j,k)

(v) s="Computer"
 for i in s[2:6]:
 print(s[3:7])

(vi) s="12345678"
 for i in s[2:6]:
 print(s[int(i):])

```

## Converting the loops:

1. Rewrite the following code snippet using a while loop:

```

(i) for i in range(10):
 print(i)

(ii) for even in range(2,20,2):
 print(even)

(iii) for k in range(11,0,-1):
 print(10,'x',k,'=',10*k)

(iv) List=[1,4,5,15,6,7]
 for k in List:
 if k%2==0:
 print(List[k%6])
 else: print(List[5*k])

```

```
(v) str='Amplifier'
 for i in str:
 if i in 'aeiou':
 print(i,end='@')
```

2. Rewrite each of the following code snippets using a for loop:

```
(i) i=0
 while i<10:
 if i%2==0:
 print(i)
 else: print(i-1)
 i+=1
(ii) even=2
 while even<20:
 print(even)
 even+=4
```

### Differentiate between:

- Assuming that A is a list, differentiate between:
  - A[2] and A[2:]
  - A[2:] and A[:2]
  - A[2::] and A[:2]
  - A\*3 and [A,A,A]
- Assuming that A is a tuple, differentiate between A\*3 and (A,A,A)
- Differentiate between:
  - 2 and 2,
  - a, b=2, 4 and a=b=2, 4
  - a=(2) and a=(2,)

### Writing code:

- Write Python script to enter temperature in degree Celsius and display it into Fahrenheit. The formula to convert from degree Celsius to degree Fahrenheit is:

$$^{\circ}\text{F} = \frac{9}{5} \times ^{\circ}\text{C} + 32$$

- Write Python script to input a time duration in seconds and display it in the following format:

\_\_\_\_\_Seconds = \_\_\_\_\_hr \_\_\_\_\_min \_\_\_\_\_seconds.

For example, if the user input is 4526, the output should be:

4526 seconds = 1 hr 15 min 26 seconds

- A school conducts two weekly tests of 30 marks each, and a term exam of 100 marks, and calculates the aggregate percentage of marks as follows:  
(weekly test total)/3 + 80% of term exam.  
Write Python script to input weekly test marks and term exam marks, and calculate aggregate percentage (rounded up to 2 digits after decimal) of a student. The script should then display weekly test marks, term exam marks, and aggregate marks.
- Write a script to input a number and check whether it is even or odd, and then display an appropriate message.
- Write a script which inputs two numbers from the user and checks if first number is divisible by the second or not. The script should then display an appropriate message.
- Write a script to input two numbers and find the greater.



7. Write a script to input four numbers and find the largest of these without using built-in function `max()`.
8. Write a script which inputs three numbers from the user and displays these numbers in descending order.
9. Write a program to display even natural numbers from 2 to n, where n is to be input from the user. For example, if the value of n is 13, the output should be 2 4 6 8 10 12.
10. Write a program to find the sum of first n natural numbers, where n is to be input from the user.
11. Write a program to find the sum of first n even natural numbers, where n is to be input from the user.
12. Write a program to input an integer and find its factorial. Factorial of an integer n ( $n \geq 1$ ), is defined as the product of all natural numbers from 1 to n, and is represented by n!. For example, factorial of 5 is represented by 5! and is equal to 120 ( $=1 \times 2 \times 3 \times 4 \times 5$ ). Factorial of 0 is defined as 1 and factorial of negative integers is not defined.
13. Write a program to input 10 numbers from the user and then display the largest of the numbers entered.
14. Write a program to input 10 numbers from the user and then display their sum and average. Also display the largest and the smallest of the numbers entered.
15. Write a program to input a number from the user and display its first 10 multiples.
16. Write a program to input two numbers m and n from the user. Then display first m multiples of n.
17. Write a program to input 10 numbers from the user and then display the smallest number entered. If the smallest number is an integer, then find the sum of its digits, otherwise display the message "smallest number is not an integer".
18. Write a program to input an integer and find the sum of its digits.
19. Write a program to input an integer and find the product of its odd digits. If the number does not contain any odd digit, then the program should display an appropriate message instead of the product.
20. Write a program to input 10 integers and find the sum of the digits of each integer.
21. Write a program to input an integer and check whether it is palindrome or not.
22. Write a program to input a number and check whether it is a perfect number or not. (A number is a perfect number if it is equal to the sum of its own factors, except itself. For example, 6 is a perfect number as  $6=1+2+3$ , and 8 is not a perfect number as  $8 \neq 1+2+4$ ).
23. Assuming that **nums** is a list of integers, write code to find the length of **nums** without using the inbuilt function `len()`.
24. Assuming that **num** is a list of integers, write Python code to divide all those elements of list by 5 which are divisible by 5 and multiply the other elements of the list by 2. No element of the list should be changed to float as a result of division by 5. The code should then display the original list and the changed list.
25. Assuming that **num** is a list of integers, write code display the list. The code should then replace each element having even value with its half and each element having odd value with its double, and then display the changed list.  
Example: if the list is [3, 4, 5, 16, 9], then the modified list should be [6, 2, 10, 8, 18]
26. Write a program to input 10 numbers from the user and then display this list of numbers in reverse order.
27. Write a program to create a list of 10 random integers in the range 10 to 99, and display this list. Then display all the odd elements of the list.
28. Write a program to create a list of 10 random numbers in the range 10.00 to 99.99, and display this list. Then interchange first half of the list with second half and display the updated list. For example, if the original list is:  
[98.39, 50.84, 42.57, 65.37, 46.78, 37.67, 16.98, 76.22, 54.24, 44.7]  
then after interchange, the list should become  
[37.67, 16.98, 76.22, 54.24, 44.7, 98.39, 50.84, 42.57, 65.37, 46.78]
29. Write a program to create a list of n random integers in the range 1 to 25, and display the list. Then exchange the values of first half side elements with the second half side elements of the list. n is

- to be input from the user. For example, if the generated list is [2, 4, 1, 6, 7, 9, 23, 10], then after the exchange, the list should become [7, 9, 23, 10, 2, 4, 1, 6].
30. Write a program to create a list of 10 random numbers in the range 10.00 to 99.99, and display this list. Then interchange consecutive elements of the list and display the updated list. For example, if the original list is:  
     [88.76, 41.43, 35.9, 70.06, 88.77, 53.68, 33.01, 83.42, 35.29, 98.22]  
 then after interchange, the list should become  
     [41.43, 88.76, 70.06, 35.9, 53.68, 88.77, 83.42, 33.01, 98.22, 35.29]
31. Write a function which accepts a list of integers and reverses the contents of the list without slicing the list and without using any second list. For example, if the original list is [2, 15, 3, 14, 7, 9, 19, 6, 1, 10], then after reversal the list should be [10, 1, 6, 19, 9, 7, 14, 3, 15, 2].
32. Write a function which takes a list as a parameter. The program should then shorten the list by removing all the duplicate elements from it. The original order of the elements in the list should be maintained. For example, if the original list is:  
     [16, 15, 20, 6, 9, 12, 18, 16, 2, 19, 9, 20, 20, 14, 17, 2, 19, 12, 10, 13]  
 then the shortened list should be:  
     [16, 15, 20, 6, 9, 12, 18, 2, 19, 14, 17, 10, 13]
33. Write a function which takes two lists A and B as parameters. The function should delete from list A all the elements which are present in list B too.
34. Write a function which takes two equi-sized lists A and B as parameters, and then combines their contents into a third list C by computing  $C[i] = 2 * A[i] + 3 * B[i]$ ; where value i varies from 0 to N-1, N being the size of each list. The function should then return list C.
35. Write a function which takes two lists, say **first** and **second**, as parameters and transfers the content of first and second to a third list named **ALL**. The even places in **ALL** should be filled from list **first** and odd places of **ALL** will be filled from list **second**. For example, if **first** is [1, 2, 3, 4, 5, 6] and **second** is [10, 20, 30, 40], then **resultant ALL** should be [10, 1, 20, 2, 30, 3, 40, 4, 5, 6]. At the end, the function should return the list **ALL**.
36. Write a function `rotate(A)`, where A is a list. The function should re-position all elements of A by shifting each element, starting from the second position, to one position before and by shifting the first element to the last position. For example, if A is [22, 25, 70, 32, 12], then after the function execution, A should be [25, 70, 32, 12, 22].  
 Do not use slicing in the function.
37. Write a function `swap10(nums)`, where `nums` is a list. The function should modify the contents of `nums` in such a way that the elements, which are multiples of 10, swap with the value present in the very next position in the list. For example, if the original list P is [91, 50, 54, 22, 30, 54], then after modification, P should become [91, 54, 50, 22, 54, 30].
38. Write a function which takes a list as a parameter and changes all the multiples of 5 in the list to 5 and rest of the elements as 0. For example, if the list is [55, 43, 20, 16, 39, 90, 83, 40, 48, 25], then after executing the function the list should be changed to [5, 0, 5, 0, 0, 5, 0, 5, 0, 5].
39. Write a function which takes a list as a parameter and replaces each even element of the list with the sum of its digits and each odd element with the product of its digits. For example, if the list is given as [100, 43, 20, 56, 32, 91, 80, 40, 45, 21], then the function should change the list to [1, 12, 2, 11, 5, 9, 8, 4, 20, 2].
40. Write a function which takes a list, say *Pay*, as a parameter and modifies each element of *Pay* as per the following rules:

| Existing value of Pay element   | Change required in Pay element |
|---------------------------------|--------------------------------|
| If less than 100000             | Increase existing value by 25% |
| If $\geq 100000$ and $< 200000$ | Increase existing value by 20% |
| If $\geq 200000$                | Increase existing value by 15% |

41. Write a function which combines two tuples of strings into a third tuple and sorts this resultant tuple in ascending order. The function should then return this sorted tuple.
42. Assume that **nums** is a tuple in Python and **ele** is an element to be searched for in the list. Write a function to search for **ele** in **nums** without using any inbuilt method of tuple class. The function should return the position (index) where the **ele** is present in **nums**. If **ele** is not present in **nums**, the function should return -1.
43. Write a function intersection(A,B), which takes two tuples, A and B, as parameters. The function should create another tuple which contains common elements of these two tuples, and return this new tuple. All elements in the resultant tuple should be unique. For example, If A=(1,2,3,'a','b',3,1) and B=(3,4,8,'a'), then the resultant tuple should be (3,'a').
44. Write a function which takes a tuple of integers as a parameter. The function should then create another tuple named NEW\_NUMBERS in such a way that NEW\_NUMBERS contains the elements from the alternate locations of NUMBERS starting from 0, and return NEW\_NUMBERS. For example, if NUMBERS is (2, 5, 9, 14, 17, 8, 19, 16), then NEW\_NUMBERS should be (2, 9, 17, 19)
45. Write a function which takes two equi-sized tuples A and B as parameters, and then combines their contents into a third tuple C by computing  $C[i]=2*A[i]+3*B[i]$ ; where value i varies from 0 to N-1, N being the size of each list. The function should then return tuple C.
46. Write a program to create a dictionary *employee* with the keys:
  - a. **name**: Name of the employee
  - b. **basic**: basic salary of the employee (an integer)
  - c. **dept**: Department (A string)

Then add the following keys to the dictionary:

- d. **hra**: House Rent Allowance (float – 40% of the basic)
- e. **da**: Dearness Allowance (float – 50% of the basic)
- f. **ta**: float – to be calculated as follows:

| basic                         | ta               |
|-------------------------------|------------------|
| If less than 10000            | 10% of the basic |
| If $\geq 10000$ and $< 20000$ | 20% of the basic |
| If $\geq 20000$               | 30% of the basic |

- g. **gross**: Gross salary (float – basic+hra+da+ta)
  - h. **pf**: Provident Fund (float – 8.5% of the basic)
  - i. **net**: Net salary (float – Gross salary-Provident Fund)
- Input the values of name, basic, and dept from the user and calculate the values of other keys. Then display all the elements of *employee* in different lines.
47. Write a function which takes a string as a parameter and displays it in reverse order without using slicing.
  48. Write a function which takes a string as a parameter. The function should count and display the number of lowercase alphabets, uppercase alphabets, and digits in the string.
  49. Write a function which takes two strings as parameters and creates a new string by joining first three characters of the first string and last four characters of the second string. Assume that each string has sufficient number of characters. The function should then return this resultant string.
  50. Write a function which takes a string, say s, as a parameter and creates a new string by removing all the duplicate characters from s. For example, if the s is *Mississippi*, then the new string should be *Misp*. The function should then return this resultant string.
  51. Write a function which takes the first name, middle name, and last name of a person as parameters. The function should create and return a string containing the short name of the person. For example if the person's name is Chander Mohan Subramanian, then the function should return C.M.Subramanian.
  52. Write a function which takes a string as a parameter and checks whether it is a palindrome or not. The function should return True if the string is palindrome, otherwise it should return False. The checking should not be case-sensitive.

53. Write a function which takes a string as a parameter and returns the number of words in in the string.
54. Write a function which takes a string as a parameter and then display the string in the reverse order.  
For example, if the string is:  
    Proper and good **education** is very **important** for all of us.  
Then the displayed string should be:  
    us. of all for important very is education good and Proper
55. Write a function which takes a string as a parameter and then display the string in such a way that each word of the string is reversed. For example, if the entered paragraph is:  
    Proper and good **education** is very **important** for all of us.  
Then the displayed string should be:  
    reporP dna doog noitacude si yrev tnatropmi rof lla fo .su

# ANSWERS

## Descriptive Questions:

1. `print()` is a function to display the specified content on screen.
2. The keyword arguments of `print()` are: `end`, `file`, `flush`, `sep`.
3. Data type refers to the type of value of a data object. Fundamental data types in Python are: `bool`, `complex`, `float`, `int`, `str`.
4. (i) `int` (ii) `str` (iii) `str` (iv) `float` (v) `str` (vi) `float` (vii) `int` (viii) `float` (ix) `str` (x) `str`
5. (i) `type(12)` (ii) `type('Hello')` (iii) `type(Hello)`  
(iv) `id(address)` (v) `print('-*-'*25)`
6. An operator which takes one operand is called a unary operator. Examples of unary operators are unary `+` and unary `-`. An operator which takes two operands is called a binary operator. Examples of binary operators are `*` and `/`.
7. Operator precedence of arithmetic operators in Python is as follows:  
(i) `(**)` > (unary `+`, unary `-`) > `(*, /, //, %)` > (binary `+`, binary `-`)
8. (i) 122.0 (ii) 11.0 (iii) 14.0 (iv) 49 (v) 4 (vi) 9
9. A variable is the name given to a data object stored in memory. Rules for forming valid variable names in Python:
  - A variable name must start with an alphabet or an underscore (`_`).
  - A variable name can consist of UNICODE alphabets, digits, and underscore(`_`). No other character is allowed.
  - A Python keyword cannot be used as a variable name.
10. `id()`
11. `import` keyword  
`keyword.kwlist`
12. Python is called a dynamic type language because in Python a variable does not have any fixed data type and its data type keeps changing depending upon the value it represents.
13. An Augmented Assignment operator represents a combination of a binary operation and assignment. For example, the augmented assignment operator `+=` is a combination of binary `+` and `=`. It evaluates R-value and adds it to L-value. The final result is assigned to L-value.
- 14.

| Invalid variable name    | Reason                                                                         |
|--------------------------|--------------------------------------------------------------------------------|
| <code>if</code>          | <code>if</code> is a keyword, and a keyword cannot be used as a variable name. |
| <code>total marks</code> | A variable name cannot have a space.                                           |
| <code>S.I.</code>        | A variable name cannot have a dot ( <code>.</code> )                           |
| <code>9t</code>          | A variable name cannot start with a digit                                      |
| <code>tag\$</code>       | A variable name cannot have <code>\$</code>                                    |

15.  
(a)  $(a+b)/2$  (b)  $(3*3+9**3)/2$  (c)  $3*3+9**3/5$   
(d)  $a**0.5+(a+2)/b$  (e)  $8-6+(6*sum)/7-var**0.5$  (f)  $u*t+a*t*t/2$
16.  
(a)  $63//29$  (b)  $63\%29$  (c)  $3**(1/4)$   
(d)  $6*5+6\%5$  (e)  $(8+4**3)**0.5$  (f)  $8**0.5+(4**3)$
17. Comments are statements in a script which are ignored by the Python interpreter, and therefore, have no effect on the actual output of the code.  
A *comment* in Python starts with a `#` anywhere in a line and extends till the end of the line. The following code contains two comments:

```
This is a comment line
n=1
m=n+1 # value of n+1 is assigned to m
print(m)
```
18. **Syntax error:** A syntax error is a grammatical error in a script. Example: mismatching parentheses.  
**Run time error:** A run time error is an error which occurs during program execution due to incorrect or illegal values used in some operations. Example: division by zero.

**Logical errors:** These are the errors in the logic of the script. Example: incorrect sequence of statements in the script.

19. A **token** is the smallest meaningful element of a **Python script**. Different categories of tokens are:

| Category    | Examples   |
|-------------|------------|
| Identifiers | print, sum |
| Keywords    | if, and    |
| Literals    | 56, "abc"  |
| Operators   | +, -       |
| Delimiters  | (, )       |

- 20.

| Token | Category             | Token         | Category  |
|-------|----------------------|---------------|-----------|
| name  | Identifier           | ,             | Delimiter |
| age   | Identifier           | (             | Delimiter |
| eval  | Identifier           | )             | Delimiter |
| input | identifier           | "Arjun"       | Literal   |
| print | identifier           | "Enter age: " | Literal   |
| =     | Delimiter / Operator | "is"          | Literal   |
| "     | Delimiter            | "years old"   | Literal   |

21. (i) False (ii) True (iii) True (iv) False (v) False  
(vi) False (vii) False (viii) True (ix) False (x) True (xi) False (xii) True
22. Relational operators are the operators which are used to compare two values. Different relational operators available in Python are: ==, !=, >, >=, <, <=
23. Logical operators are the operators which are used to make complex conditions by combining and/or negating simple conditions. Different logical operators available in Python are: **not**, **and**, **or**

- 24.

- (i) num1==num2  
(ii) num1 != num2  
(iii) num2%num1==0  
(iv) num1%num2==0  
(v) num1%2==0 and num2%2==0  
(vi) num1%2!=0 and num2%2!=0  
(vii) num1%2==0 and num2%2!=0 or num1%2==0 and num2%2!=0  
(OR (num1+num2)%2 != 0)  
(viii) (num1+num2)>num1\*num2  
(ix) num1%3==0 or num1%5==0  
(x) num1%3==0 and num1%5==0 (OR num1%15==0)

25. **else** clause of a loop gets executed after the normal completion of the loop. **else** clause is not mandatory in a loop.

26. The **break** statement prematurely ends execution of the current **while** or **for** loop. It brings the control to the statement immediately following the current control structure, skipping the optional "else" clause if the loop has one.

The **continue** statement is used to skip the execution of the current iteration of a loop, and continue with the next. **continue** does not terminate the loop, but continues with the next iteration of the loop. **Example:**

```
#break and continue
for i in range(1,11):
 if (i%3 == 0):
 continue;
 if (i*3 > 25):
 break;
 print(i,end=' ')
else: print("Else part of loop")
print()
print(i);
```

Output

|             |
|-------------|
| 1 2 4 5 7 8 |
| 10          |

27. A loop which never terminates is called an infinite loop. Example:

```
i=1
while i<5:
 print(i)
```

28.

- (i) fruit=['banana','mango','apple','peach','grapes']
- (ii) City=['Delhi','Chennai','Kannur','Jalandhar','Hyderabad']
- (iii) Mixed=['banana','apple','grapes',12,13]
- (iv) Box=[1,5,2,['a','b']]
- (v) R10=[random.randint(10,50) for i in range(10)]
- (vi) F10=[random.randint(1000,4999)/100 for i in range(10)]
- (vii) y=[x\*x+2\*x-1 for x in range(-4,6)]
- (viii) C=[random.randint(-30,-20) for x in range(10)]
- (ix) A=[3,5,6.7]  
B=[A,1,2,3]
- (x) A=[3,5,6.7]  
B=[A,[1,2,3]]

29.

- (i) L1[-1]=L2
- (ii) L1[-1:]=L2

30.

- (i) True    (ii) False    (iii) False    (iv) True
- (v) True    (vi) False    (vii) False    (viii) True
- (ix) True    (x) False

31.

| Mutable Data Objects                                                       | Immutable Data Objects                                                           |
|----------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Elements of a mutable data object can be changed during program execution. | Elements of an immutable data object cannot be changed during program execution. |
| Example: List                                                              | Example: Tuple                                                                   |

32.

count() and index()

33.

- (i) fruit=('banana','mango','apple','peach','grapes')
- (ii) City=('Delhi','Chennai','Kannur','Jalandhar','Hyderabad')
- (iii) Mixed=('banana','apple','grapes',12,13)
- (iv) Box=(1,5,2,['a','b'])
- (v) R10=(random.randint(10,50) for i in range(10))
- (vi) F10=(random.randint(1000,4999)/100 for i in range(10))
- (vii) y=(x\*x+2\*x-1 for x in range(-4,6))
- (viii) C=(random.randint(-30,-20) for x in range(10))
- (ix) A=[3,5,6.7]  
B=(A,1,2,3)
- (x) A=(3,5,6.7)  
B=(A,(1,2,3))

34.

- (i) True    (ii) False    (iii) False    (iv) True
- (v) True    (vi) False    (vii) False    (viii) True
- (ix) True    (x) False

35.

- (i) {'banana':70,'mango':60,'apple':120,'peach':120,'grapes':100}
- (ii) {'Amar':80, 'Akbar':78.5, 'Anthony':80.6}
- (iii) Student={'admno': '12332', 'name': 'Rohan', 'dob': '12102006', 'class':8}
- (iv) item={'code':233, 'name': 'Soap', 'price': 28.50, 'qty':8.0}
- (v) mobile={'make':233, 'model': 'iPhone 11 Pro', 'year': 2019, 'price':115000}
- (vi) Eng2Lang={'book': 'The Great Gatsby', 'this': 'The Great Gatsby', 'year': '1925', 'price': '10', 'who': 'F. Scott Fitzgerald'}
- (vii) {'A':(0,0), 'B':(3.5,0), 'C':(4.0,5.2), 'D':(0,3.2)}

36.

- (i) True    (ii) False    (iii) False    (iv) False  
(v) False    (vi) True
37. A string is a sequence of characters enclosed within quotation marks.  
Two types of strings are supported in Python. These are – Single Line string and Multi Line string.  
A Single Line string is a sequence of characters enclosed within a pair of quotation marks. Example:  
`'Book'`  
A multiline string is a sequence of characters enclosed within a pair of three quotation marks. A multiline string may span more than one line. Example:  
`''' An example of  
A multi-line  
string'''`
38. (i) False    (ii) True    (iii) False    (iv) True  
(v) True    (vi) False    (vii) False    (viii) True

## Correcting the code:

1.
  - (i) The quotation marks of string are not properly paired. The corresponding correct statement is:  
`print("three")` ( OR `print('three')`)
  - (ii) P of Print should be in lower case. The corresponding correct statement is: `print(3)`
  - (iii) Multiple arguments of print should be separated by comma. The corresponding correct statements are: `print(44, '+', 56)` and `print(44+56)`
  - (iv) There should be comma between 2 and sep. The correct statement is: `print(3, 2, sep=': ')`
  - (v) The argument of print should be within parentheses. The corresponding correct statement is:  
`print("wisdom")`
  - (vi) The argument of print should be within parentheses. The corresponding correct statement is:  
`print('33+44')`
  - (vii) PRINT should be in lowercase. The corresponding correct statement is:  
`print("15 August 1947")`
2.
  - (i) Argument must be a string. Correct statement:  
`input("Enter a number: ")`
  - (ii) No error
  - (iii) No error
  - (iv) String argument must represent a valid integer. Correct statement:  
`int("56")` #OR any other valid integer as a string or as an integer  
OR  
The argument should be a numeric expression (not a string). Correct statement is:  
`int(56.6)`
  - (v) Argument must represent a valid integer. Correct statement:  
`int("56")` #OR any other valid integer as a string or as an integer
  - (vi) Argument must represent a valid integer. Correct statement:  
`int("56")` #OR any other valid integer as a string or as an integer
  - (vii) Argument must represent a valid integer. Correct statement:  
`int(56)` #OR any other valid integer as a string or as an integer
  - (viii) No error
  - (ix) No error
  - (x) No error
  - (xi) No error
  - (xii) String argument must represent a valid real number. Correct statement:  
`float("5.4")` #OR any other valid real number as a string  
OR  
The argument should be a numeric expression (not a string). Correct statement is:



- `int(5/4)`
- (xiii) The argument must be a string. Correct statement:  
`eval('10')`
- (xiv) No error
- (xv) No error
- (xvi) The argument must be a string. Correct statement:  
`eval('10*2')`
- (xvii) No error
- (xviii) The argument of `eval()` must be a string. Correct statement:  
`eval('int(10/3)')`
- (xix) The string argument of `int()` must represent a valid integer, or it should be a number. The argument of `eval()` must be a string. Correct statement:  
`eval("int(10/3)")`
- (xx) Similar to (xix). Correct statement:  
`eval("float(10/3)")`

3.

- (i)
- ```
n1,n2=5,4
if n1==n2:
    print("Equal")
```
- (ii)
- ```
n1=n2=12/5
if n1!=n2:
 print("Not equal" #Indentation
else: print("Equal"
```
- (iii) **age=12**
- ```
if age<=10:
    print("Primary")
elif age<=13:
    print("Middle")
elif age<=15:
    print("Secondary")
elif age<=17:
    print("Sr Secondary")
```
- (iv)
- ```
num=input("Enter an integer: ")
if num<0:
 num=-num;
if num>=0 and num<=9:
 print("Single digit number")
elif num>=10 or num<=99:
 print("Two digit number")
elif num<=999:
 print("Three digit number")
else: print("Number contains more than 3 digits")
```
- (v)
- ```
a,b,c=12,13,14
if (a>=b or a<=c and a+c<=b):
    print("A") #Indentation
else: print(a+b+c)
```

4.

- (i) `#display "Hello" 10 times`
`for i in range(1, 1011):` OR `range(1,10):`
`print("Hello")`
- (ii) `#To print first 10 natural numbers`

```

i=1
while (i<=10):
    print(i)
    i+=1
(iii) #To display first 10 even natural numbers
num=2
while (num<=20):
    print(num)
    num+=2      #Indentation

(iv) #To display first 10 odd natural numbers
for i in range (10):
    print(2*i+1)

(v) #To calculate DA as 10% of basic salary for 10 employees
for i in range(10):
    basic=input("Enter basic salary")
    da=10*basic
    print("DA =",da)

```

5.

```

(i) a=int(input("Enter a number: "))
c=1
for b in range (a,a*10.0,a):
    c*=b
    print(a,b,c)
(ii) a=int(input("Enter a number: "))
b=int(input("Enter another number: "))
start,end=a,b
if a>b:
    start,end=b,a
i=start
while i<=end:
    if (i%start==0):
        print(i,start,end='--')
    i+=1
    print(i)
(iii) a=int(input("Enter a number: "))
b=int(input("Enter another number: "))
if a>b: step=-1
else: step=1
while a!=b:
    if (a+b%2>0):
        print(a+b,sep="--")
    a+=step
(iv) a=int(input("Enter a number: "))
while a%5!=0:
    if (a%2>0):
        print(a,sep="--")
    a+=1
else: print(a)
(v) x=123045
while x%10:
    x//=10
    print(x,sep="--")
else: print(a) #Indentation
(vi) x=123045

```

```

while x%10:
    d=x%10
    for i in range(d):
        print(i)
    else: print (d)      #Indentation
    print(x,sep='--')
    x/=10
else: print(a)

```

6.

```

(i) a=[1,2,3,2,1]
    for x in a:
        print(x)
(ii) a=["Umang","Usman","Utkarsh","Umedh"]
     for x in range(len(a)):
         print(a[x])
(iii) A=8*[3]
      for i in range(5):
          A[i]=eval("A[i]*i")      OR A[i]=A[i]*i
      for i in range(len(A)):      OR for i in A:
          print(A[i])              OR print(i)

```

7.

```

(i) a=(1,2,3,2,1)
    for x in a:
        print(x)

(ii) a=("Umang","Usman","Utkarsh","Umedh")
     for x in range(len(a)):
         print(a[x])

```

8.

```

(i) A="Rajasthan"
    #A[3]='a' - strings are immutable.
    for i in A:
        print(i)

(ii) s="Story teller"
     if (s[0].isupper()):
         print("s starts with upper case letter")

```

Finding the Output:

1. (i) Hello (ii) 5 -33 (iii) 2+3 5 (iv) 2+3=5 (v) 8*9 (vi) 8 and 0.125 and -8 and -8

2.

```

(a) 8 5      (b) 0 2      (c) 20 300
(d) 66.67    (e) 4 -10 7   (f) 8 8 0
    126.67

```

3. (i) 56 (ii) 56 (iii) 1 (iv) 12.3 (v) -12.45 (vi) 5.0 (vii) 10 (viii) 12 (ix) 3

4.

```

(i) Leap year
(ii) Non integer
(iii) Not eligible to vote
(iv) (a) Two digit number (b) Single digit number
      (c) Two digit number (d) Two digit number
(v) B

```

5.

```

(i) 4 4 4

```

4 8 12
4 12 24

(ii) 4--8--11 # for a=4, b=10
-4--0--3 # for a=2, b=-4

(iii) 14
13
12
11
10
9

(iv) 10 # for a=10
-5 # for a=-6

(v) 12304
1230
1230

(vi) 0
1
2
3
4
5
123045
0
1
2
3
4
12304

6.

(i) 2 2
3 3
3 2
(ii) 0 2
1 2
2 2*2 2
3 3*3 2*3 2
3 2
(iii) 0
1 2
2
2 2

7.

(i) 12
(iv) [8, 10, 12, 23, 33, 16, 6, 2, 1]
(vii) [12, 23, 33, 16, 6, 2, 1]
(x) [1, 16, 12, 7]
(xiii) 9

(ii) 8
(v) [6, 2, 1]
(viii) [8, 6]
(xi) [8, 23]
(xiv) 3

(iii) 6
(vi) [3, 8, 23, 6]
(ix) [6, 33, 12, 8, 6]
(xii) []
(xv) 4

8. 2*3*2*3*2*

9. 0
0
0

[2, 4] [-2, -3, 3, -4]

4 -2
[2] [-3, 3, -4]

10. 1 appears in nums 3 times. Its first occurrence is at index 0
2 appears in nums 3 times. Its first occurrence is at index 1
4 appears in nums 1 times. Its first occurrence is at index 7
5 appears in nums 2 times. Its first occurrence is at index 4
7 appears in nums 2 times. Its first occurrence is at index 2
9 appears in nums 1 times. Its first occurrence is at index 3

- 11.
- (i) [5, 3, 8]
 - (ii) [[5, 3, 8], 8, 66, 45, 'A', ['A', [5, 3, 8]], ['nested', 'lists']]
 - (iii) 5
 - (iv) [5, 3, 8]
 - (v) []
 - (vi) [5, 3, 8]
 - (vii) 3
 - (viii) [3, 8]
 - (ix) A
 - (x) ['A', [5, 3, 8]]
 - (xi) [5, 3, 8]*8*66*45*A*['A', [5, 3, 8]]*['nested', 'lists']*
 - (xii) ['A', [5, 3, 8]]
45
66

12. [1, 2, 3, 4, 5, 7, 9]
[5, 3, 9, 7, 1, 2, 4]
[5, 3, 9, 7, 1, 2, 4]
[5, 3, 9, 7, 1, 2, 4]

13. [1, 2, 7]
[5, 3, 9, [1, 2, 7], 4]
[5, 3, 9, [1, 2, 7], 4]
[5, 3, 9, [1, 2, 7], 4]
[5, 3, 9, [7, 1, 2], 4]
[1, 10, 7]
[5, 3, 9, [1, 10, 7], 4]
[5, 3, 9, [1, 10, 7], 4]
[5, 3, 9, [1, 10, 7], 4]
[5, 3, 9, [7, 55, 2], 4]

- 14.
- | | | |
|--|---------------|------------------------|
| (i) 12 | (ii) 8 | (iii) 6 |
| (iv) (8, 10, 12, 23, 33, 16, 6, 2, 1) | (v) (6, 2, 1) | (vi) (3, 8, 23, 6) |
| (vii) (8, 10, 12, 23, 33, 16, 6, 2, 1) | (viii) (8, 6) | (ix) (6, 33, 12, 8, 6) |
| (x) (1, 16, 12, 7) | (xi) (8, 23) | (xii) () |
| (xiii) 9 | (xiv) 3 | (xv) 4 |

15. 2#3#2#3#2#

- 16.
- 0
2
4
6
8

```

3
2
(2, 3, 4, 5, 6) (-2, -1, 3, 0, 1, 2)
3 2
(2, 3, 4, 5, 6) (-2, -1, 3, 0, 1, 2)

```

17. 1 appears in nums 3 times. Its first occurrence is at index 0
 2 appears in nums 3 times. Its first occurrence is at index 1
 4 appears in nums 1 times. Its first occurrence is at index 7
 5 appears in nums 2 times. Its first occurrence is at index 4
 7 appears in nums 2 times. Its first occurrence is at index 2
 9 appears in nums 1 times. Its first occurrence is at index 3

18.

```

(i) (5, 3, 8)
(ii) ((5, 3, 8), 8, 66, 45, 'A', ['A', (5, 3, 8)], ['nested', 'tuples'])
(iii) 5 (iv) (5, 3, 8) (v) ()
(vi) (5, 3, 8) (vii) 3 (viii) (3, 8)
(ix) A (x) ['A', (5, 3, 8)]
(xi) (5, 3, 8) (xii) ['A', (5, 3, 8)]
      8
      66
      45
      A
      ['A', (5, 3, 8)]
      ['nested', 'tuples']

```

19. [1, 2, 3, 4]
 [1, 2, 3, 4]
 (1, 2, 3, 4)
 (1, 2, 3)

20. [1, 2, 3, 4]
 [1, 2, 3]
 (1, 2, 3, 4)
 (1, 2, 3)

21. 5
 Programming
 ('U5', 'Networks')
 Files
 RDBMS
 True
 False
 True
 True

22. (i) A1 - 8 (ii) A1 - 8 (iii) 8 7 6 5 4 3 2 1 0 (iv) A1 - 8
 A2 - 7 A2 - 7 A2 - 7
 B1 - 6 B1 - 6 B1 - 6
 B2 - 5 B2 - 5 B2 - 5
 C1 - 4 C1 - 4 C1 - 4

C2 - 3	C2 - 3
D1 - 2	D1 - 2
D2 - 1	D2 - 1
E - 0	E - 0

C2 - 3
D1 - 2
D2 - 1
E - 0

23.

(i) 21	(ii) s	(iii) m
(iv) sionate Programmer	(v) mer	(vi) Psnerrm
(vii) sionate Programmer	(viii) sa	(ix) magr tnisP
(x) rmgPtos	(xi) snerr	(xii)
(xiii) 18	(xiv) 3	(xv) 7

24.

(i) 5	(ii) 2	(iii) 2345 [2, 3, 4, 5]	(iv) U*i*e* Nationsa*i*n*
2	2	345 [3, 4, 5]	
1	1	45 [4, 5]	
1		5 [5]	
		[]	
(v) nmcAt	(vi) c*OMm	(vii) C*-	
	u*TEe	P*-	
		E-	

25.

(i)	6 7 7 2
	s1 is an alphabetic string
	s1 is not a numeric string
	s1 is an alphanumeric string
	s3 is not an alphabetic string
	s3 is not a numeric string
	s3 is not an alphanumeric string
	s3 is in Title case
(ii)	<class 'str'> 9 S,p,l,i,t
	<class 'list'> 5 ['S', 'p', 'l', 'i', 't']
	<class 'list'> 3 ['One,', 'Two,', 'Three']
(iii)	Eng,BSt,CompSc
	Eng,BSt,CompSc
(iv)	6 6 9
(v)	pute
	pute
	pute
	pute
(vi)	45678
	5678
	678
	78

Converting the loops:

1.

```
(i)    i=0
        while i<10:
            print(i)
            i+=1

(ii) even=2
        while even<20:
            print(even)
            even+=2

(iii) k=11
        while k>0:
            print(10,'x',k,'=',10*k)
            k-=1

(iv) List=[1,4,5,15,6,7]
        i=0
        while i<len(List):
            k=List[i]
            if k%2==0:
                print(List[k%6])
            else: print(List[5%k])
            i+=1

(v) str='Amplifier'
        k=0
        while k<len(str):
            i=str[k]
            if i in 'aeiou':
                print(i,end='@')
            k+=1
```

2.

```
(i)    for i in range(10):
        if i%2==0:
            print(i)
        else: print(i-1)

(ii)   for even in range(2,20,4):
        print(even)
```

Differentiate between:

1.

- (i) A[2] is an element of A and A[2:] is a slice of A.
- (ii) Slice A[2:] contains all elements of A starting from A[2] till the last element. Slice A[:2] contains only the elements A[0] and A[1].
- (iii) Slice A[2:] contains all elements of A starting from A[2] till the last element. Slice A[::2] contains alternate elements of A starting from A[0] till the last element.
- (iv) A*3 represents a list which contains all elements of A thrice. [A,A,A] represents a list which contains three instances of list A.

2. A*3 represents a tuple which contains all elements of A thrice. (A,A,A) represents a tuple which contains three instances of list A.

3.

- (i) 2 is an integer, and 2, is a tuple.
- (ii) a,b=2,4 assigns integer values 2 and 4 to variables a and b respectively. a=b=2,4 assigns tuple (2,4) to both a and b.
- (iii) a=(2) assigns integer value 2 to a. a=(2,) assigns tuple (2,) to a.

Writing code:

1.

```
c=float(input("Enter temperature in degree Celcius: "))
f=9/5*c+32
print("Temperature in degree Fahrenheit = ",f)
```
2.

```
s=int(input("Enter time duration in seconds: "))
t=s
h=s//3600
s=s%3600
m=s//60
s=s%60
print(t,"seconds =",h,"hours",m,"minutes",s,"seconds")
```
3.

```
wt1=float(input("Enter first weekly test marks: "))
wt2=float(input("Enter second weekly test marks: "))
te=float(input("Enter term exam marks: "))
agg=(wt1+wt2)/3+te*80/100
agg=round(agg,2)
print("First weekly test marks:",wt1)
print("Second weekly test marks:",wt2)
print("Term exam marks:",te)
print("Aggregate: ",agg,"%")
```
4.

```
#To input a number and find whether it is even or odd
n = eval(input('Enter a number: '))
if (n%2==0):
    print ("Even")
else:
    print ("Odd")
```
5.

```
a=eval(input("Enter first number: "))
b=eval(input("Enter second number: "))
if a%b==0:
    print("First number is divisible by the second")
else: print("First number is not divisible by the second")
```
6.

```
# To input two numbers and find the greater of the two
n1 = eval(input('Enter first number: '))
n2 = eval(input('Enter second number: '))
if n1>n2:
    print (n1,"is greater")
elif n2>n1:
    print (n2,"is greater")
else:
    print("numbers are equal")
```
7.

```
print("Enter four numbers, press Enter after each number: ")
a=eval(input())
b=eval(input())
c=eval(input())
d=eval(input())
if a>=b and a>=c and a>=d:
    print(a,"is the largest")
```

- ```

elif b>=a and b>=c and b>=d:
 print(b,"is the largest")
elif c>=a and c>=b and c>=d:
 print(c,"is the largest")
else: print(d,"is the largest")
else: print("The numbers are equal")

8. a=eval(input("Enter first number: "))
 b=eval(input("Enter second number: "))
 c=eval(input("Enter third number: "))
 if a>b and a>c:
 if b>c:
 print (a,b,c)
 else: print(a,c,b)
 elif b>a and b>c:
 if a>c:
 print(b,a,c)
 else: print(b,c,a)
 elif a>b:
 print(c,a,b)
 else: print(c,b,a)

9. # To display even natural numbers from 2 to n
 n = int(input("Enter the value of n: "))
 for num in range(2, n+1, 2):
 print(num)

10. # To find the sum of first n natural numbers
 n = int(input("Enter the value of n: "))
 sum=0
 for i in range(1,n+1):
 sum += i
 print("sum of first",n,"natural numbers =",sum)

11. # To find the sum of first n even natural numbers
 n = int(input("Enter the value of n: "))
 sum=0
 for num in range(2, 2*n+1, 2):
 sum += num
 print("sum of first",n,"even natural numbers =",sum)

```

**OR**

- ```

# To find the sum of first n natural numbers
n = int(input("Enter the value of n: "))
sum=0
for num in range(1, n+1):
    sum += 2*num
print("sum of first",n,"even natural numbers =",sum)

12. # To find the factorial of an integer
    n = eval(input("Enter the value of n: "))
    n=abs(int(n)) #because factorial of negative numbers is not defined
    factorial=1
    for num in range(2, n+1): #multiplication with 1 is redundant
        factorial *= num
    print(factorial)

13. # To find the largest of 10 numbers entered
    n = eval(input("Enter a number: "))
    max=n
    for i in range(1,10):

```

```

        n=eval(input("Enter a number: "))
        if n>max:
            max=n
    print(max,"is the largest number entered")

14.    sum=0
        n=Max=Min=eval(input("Enter first number: "))
        sum+=n
        for i in range(9):
            n=eval(input("Enter next number: "))
            sum+=n
            if n>Max:
                Max=n
            elif n<Min:
                Min=n
        avg=sum/10
        print("Sum =",sum)
        print("Average =",avg)
        print("Largest number entered =",Max)
        print("Smallest number entered =",Min)

15.    n = eval(input("Enter a number: "))
        for i in range(1,11):
            print(n,"x",i,"=",n*i)

16.    m=int(input("Enter the value of m: "))
        n=eval(input("Enter the value of n: "))
        print("First",m,"multiples of",n,"are: ")
        for i in range (1,m+1):
            print(n,"x",i,"=",n*i)

17.    n = eval(input("Enter a number: "))
        min=n
        for i in range(1,10):
            n=eval(input("Enter a number: "))
            if n<min:
                min=n
        print(min,"is the smallest number entered")
        if (type(min)==int):
            min=abs(min) # to make the number positive, if it is negative
            sum=0
            while min != 0:
                digit = min%10
                sum += digit
                min//=10
            print ("Sum of the digits of smallest number",sum)
        else: print("Smallest number is not an integer")

18.    n = int(input("Enter an integer: "))
        sum=0
        while n != 0:
            digit = n%10
            sum += digit
            n//=10
        print (sum)

19.    # To input an integer and find the product of its odd digits.
        n = int(input("Enter a number: "))
        n = abs(int(n))
        prod, Found=1, False
        while n != 0:

```

```

        digit = n%10
        if digit%2 == 1:
            prod *= digit
            Found=True
        n //= 10
    if Found: # OR Found==True
        print("Product of odd digits =",prod)
    else:
        print("No odd digit is present in the number")

20. #To find the sum of the digits of an integer
    for i in range (10):
        n = int(input("Enter an integer: "))
        n=abs(n)
        sum=0
        while n != 0:
            digit = n%10
            sum += digit
            n//=10
        print (sum)

21. n=int(input("Enter the number: "))
    temp=n
    rev=0
    while n>0:
        d=n%10
        rev=rev*10+d
        n//=10
    if rev==temp:
        print(n,"is a Palindromic number")
    else: print(n,"is not a Palindromic number")

22. n=int(input("Enter the number: "))
    sum_of_factors=0
    for i in range(1,n//2+1):
        if n%i==0:
            sum_of_factors+=i
    if n==sum_of_factors:
        print(n,"is a perfect number")
    else: print(n,"is not a perfect number")

23. n=0 #The counter
    for i in nums:
        n+=1
    print("length of the list is: ",n)

24. print("The original list is: ",num)
    for i in range(len(num)):
        if num[i]%5==0:
            num[i]//=5
        else: num[i]*=2
    print("The changed list is: ",num)

25. print("The original list is: ",num)
    for i in range(len(num)):
        if num[i]%2==0:
            num[i]//=2
        else: num[i]*=2
    print("The changed list is: ",num)

26. num=[] #[] creates an empty list
    for i in range(10):

```

```

        n=int(input("Enter a number: "))
        num+=[n] #Inserts number n in the list
        print("List in reverse order is: ")
        for i in range(-1,-11,-1): #OR range(9,-1,-1)
            print(num[i],end=' ')
27.    from random import randint
        num=[randint(10,99) for i in range (10)] #[] creates the list
        print("The list is: ",num)
        print("Odd elements of the list are: ", end=" ")
        for i in num:
            if (i%2==1):
                print(i, end=" ")
28.    from random import random
        num=[ round(random()*100,2) for i in range (10)]
        print("The list is: ",num)

        i,j=0,5 #first half of the list starts at num[0], and
                #second half starts at num[5]
        while i<5:
            num[i],num[j]=num[j],num[i]
            i+=1
            j+=1
        print("The list after swap is: ",num)
29.    from random import randint
        n=int(input("How many numbers do you want in the list? "))
        num=[ randint(1,25) for i in range (n)]
        print("The list is: ",num)

        i,j=0,(n+1)//2 #first half of the list starts at num[0], and
                        #second half starts at num[(n+1)//2]
        while j<n:
            num[i],num[j]=num[j],num[i]
            i+=1
            j+=1
        print("The list after swap is: ",num)

30.    from random import random
        num=[round(random()*100,2) for i in range(10)]
        print("The list is: ",num)
        for i in range(0,10,2):
            num[i],num[i+1]=num[i+1],num[i]
        print("The list after swap is: ",num)
31.    def reverse_list(NUMBERS):
        i,j=0,len(NUMBERS)-1
        while i<j:
            NUMBERS[i],NUMBERS[j]=NUMBERS[j],NUMBERS[i]
            i+=1
            j-=1

32.    def shorten_list(nums):
        nums.reverse()
        n=len(nums)
        for ele in nums:
            while nums.count(ele)>1:
                nums.remove(ele)
        nums.reverse()

33.    def A_minus_B(A,B):
        for ele in A:
            if ele in B:

```

```
A.remove(ele)
```

34. `def ABC(A,B):`
 `C=[]`
 `for i in range(len(A)):`
 `C.append(2*A[i]+3*B[i])`
 `return C`
35. `def to_ALL(first, second):`
 `ALL=[]`
 `l1=len(first)`
 `l2=len(second)`
 `i=j=0`
 `while i<l1 and j<l2:`
 `ALL.append(second[j])`
 `ALL.append(first[i])`
 `i+=1`
 `j+=1`
 `ALL.extend(first[i:])`
 `ALL.extend(second[j:])`
 `return ALL`
36. `def rotate(A):`
 `n=len(A)`
 `temp=A[0]`
 `for i in range(n-1):`
 `A[i]=A[i+1]`
 `A[n-1]=temp`
37. `def swap10(nums):`
 `i=0`
 `while i<len(nums)-1:`
 `if nums[i]%10==0:`
 `nums[i],nums[i+1]=nums[i+1],nums[i]`
 `i+=2`
 `else:`
 `i+=1`
38. `def multiple5(P):`
 `for i in range(len(P)):`
 `if P[i]%5==0:`
 `P[i]=5`
 `else:`
 `P[i]=0`
39. `def sum_prod(P):`
 `l=len(P)`
 `for i in range(l):`
 `num=P[i]`
 `s,p=0,1`
 `while num:`
 `s+=num%10`
 `p*=num%10`
 `num//=10`
 `if P[i]%2==0:`
 `P[i]=s`
 `else:P[i]=pl=len(P)`

```

40. def modify_pay(Pay):
    l=len(Pay)
    for i in range(l):
        if Pay[i]<100000:
            Pay[i]+=0.25*Pay[i]
        elif Pay[i]<200000:
            Pay[i]+=0.20*Pay[i]
        else:
            Pay[i]+=0.15*Pay[i]

41. n1=int(input("Enter number of students in first class: "))
    def merge(t1,t2):
        t3=t1+t2
        t3=sorted(t3)
        t3=tuple(t3)
        return t3

42. def search(nums,ele):
    n=len(nums)
    found=-1
    for i in range(n):
        if ele==nums[i]:
            found=i
            break
    return found

43. def intersection(t1,t2):
    t3=()
    for i in t1:
        if i in t2 and i not in t3:
            t3+=(i,)
    return t3

44. def alternate(NUMBERS):
    n=len(NUMBERS)
    NEW_NUMBERS=()
    for i in range(0,n,2):
        NEW_NUMBERS+=NUMBERS[i],
    return NEW_NUMBERS

45. def ABC(A,B):
    C=()
    for i in range(len(A)):
        ele=2*A[i]+3*B[i]
        C+=(ele,)
    return C

46. name=input("Enter name: ")
    basic=int(input("Enter basic salary: "))
    dept=input("Enter department: ")
    employee={'name':name, 'basic':basic, 'dept':dept}
    hra=.40*basic
    da=.5*basic
    if basic<10000:
        ta=.1*basic
    elif basic<2000:
        ta=.2*basic
    else: ta=.3*basic
    gross=basic+hra+da+ta
    pf=.085*basic

```

```

net=gross-pf
employee['hra']=hra
employee['da']=da
employee['ta']=ta
employee['gross']=gross
employee['pf']=pf
employee['net']=net
for k,v in employee.items():
    print(k,":",v)

47.  def str_reverse(s):
        n=len(s)
        for i in range(-1,-n-1,-1):
            print(s[i],end='')

48.  def count(s):
        l=len(s)
        lc=uc=d=0
        for ch in s:
            if ch.islower():
                lc+=1
            elif ch.isupper():
                uc+=1
            elif ch.isdigit():
                d+=1
        print("Number of lowercase alphabets:",lc)
        print("Number of uppercase alphabets:",uc)
        print("Number of digits:",d)

49.  def first_last(s):
        ns=f[:3]+s[-3:]
        return ns

50.  def del_dup(s):
        s1=''
        for ch in s:
            if ch not in s1:
                s1+=ch
        return s1

51.  def name(f,m,l):
        fn=f[0]+"."
        if len(m)>0:
            fn += m[0]+"."
        fn += l
        return fn

52.  def palindrome(s):
        s=s.lower()
        rev=s[::-1]
        return s==rev

53.  def count_words(p):
        s=p.split()
        w=len(s)
        return w

```



```
54. def reverse1(p):  
    words=p.split()  
    for w in words[::-1]:  
        print(w,end=' ')  
  
55. def reverse2(p):  
    words=p.split()  
    for w in words:  
        print(w[::-1],end=' ')
```