Q1. We have a list = [1,2,3,4], modified such that the new values stored in the list are: list = [1,2,10,4,5,6]. What were the possible operations used to make the changes?

1. list[2:3]=[10]

list[3:]=[5,6]

1. list[2:2]=[10]

list[3:]=[5,6]

1. list[2:2]=[10]

list[3:5]=[5,6]

1. list.insert(2,10)

list.delete(2)

list[3:]=[5,6]

ANS: A

Q2.num=19

string=’my age is’

Which option will give the correct output?

Output: my age in 19

1. print string+ str(num)
2. print string+`num`
3. print string + num
4. print string and num
5. A and B
6. A and D
7. A, B and C
8. A, B and D

ANS: E

Q3. Monty needs to get the secret message by removing all the “X” from the string s = ”XXpyXXtXXhXoXXXnX”

Simple isn’t it?

The catch is , monty needs to write a python code to extract the message.

Which of these codes will help him decipher the string?

A. print filter(lambda x:x!=’X’,s)

B. print filter(lambda x:x==’X’,s)

C. print filter(lambda x:x!=’X’ and x%2==0,s)

D. print filter(lambda x:x,s)

ANS: A

Q4. Balancing of brackets:

A bracket is considered to be any one of the following characters: (or).

Two brackets are considered to be a *matched pair* if an opening bracket i.e. (occurs to the left of a closing bracket i.e.) *of the exact same type*. A matching pair of brackets is *not balanced* if the set of brackets it encloses are not matched.

E.g. (d+(c\*(a+b)/e) is not balanced

E.g.((a-b) \*c/d) is balanced

By this logic, we say a sequence of brackets is considered to be *balanced* if the following conditions are met:

* It contains no unmatched brackets.
* The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.

By this logic, which of the following is the correct code for checking whether the expression has balanced brackets?

A.

stri='(()))('

c=0

for i in stri :

if i == '(':

c=c+1

if i == ')':

c=c-1

if c > 0 :

print ('not balanced')

exit(0)

if c != 0 :

print ("not balanced")

else :

print ("balanced")

B.

stri='(()))('

c=0

for i in stri :

if i == '(':

c=c+1

if i == ')':

c=c-1

if c < 0 :

print ('not balanced')

exit(0)

if c != 0 :

print ("not balanced")

else :

print ("balanced")

C.

stri='(()))('

c=1

for i in stri :

if i == '(':

c=c+1

if i == ')':

c=c-1

if c < 0 :

print ('not balanced')

exit(0)

if c != 0 :

print ("not balanced")

else :

print ("balanced")

D.

stri='(()))('

c=0

for i in stri :

if i == '(':

c=c+1

if i == ')':

c=c+1

if c < 0 :

print ('not balanced')

exit(0)

if c != 0 :

print ("not balanced")

else :

print ("balanced")

ANS:B