Top of Form

Question 1: **Incorrect**

**You are developing a python application for your company.**

**A list named employees contains 500 employee names,the last 3 being company management. Which of the following  represents only management employees.**

* ​

employees[497:500]

* ​

employees[-3:]

* ​

employees[497:]

**(Incorrect)**

* ​

All the above

**(Correct)**

**Explanation**

list[begin:end] returns list of elements from begin index to end-1 index default value for begin is: 0

Bottom of Form

Top of Form

Question 2: **Correct**

**Which of the following expression will generate max value?**

* ​

8//3\*4

* ​

8%3\*4

* ​

8-3\*4

* ​

8/3\*4

**(Correct)**

**Explanation**

8%3\*4--->8  
8-3\*4 -->-4  
8//3\*4--->8  
8/3\*4--->10.6666

Bottom of Form

Top of Form

Question 3: **Correct**

**In which of the following cases we will get same result**

* ​

11/3

* ​

23%5

**(Correct)**

* ​

3\*\*1

**(Correct)**

* ​

13//4

**(Correct)**

**Explanation**

23%5-->3  
3\*\*1--->3  
11/3--->3.6666  
13//4--->3

Bottom of Form

Top of Form

Question 4: **Correct**

**Which of the following code snippet will produce the output:**

1. **Boy**
2. **Cat**
3. **Dog**

* ​
  1. l=['Apple','Boy','Cat','Dog']
  2. for x in l:
  3. if len(x) == 3:
  4. print(x)

**(Correct)**

* ​
  1. l=['Apple','Boy','Cat','Dog']
  2. for x in l:
  3. print(x)
* ​
  1. l=['Apple','Boy','Cat','Dog']
  2. for x in l:
  3. if len(x) != 3:
  4. print(x)
* ​
  1. l=['Apple','Boy','Cat','Dog']
  2. l1=l[1:]
  3. for x in l1:
  4. print(x)

**(Correct)**

**Explanation**

l=['Apple','Boy','Cat','Dog']  
for x in l:  
if len(x) == 3:  
print(x)  
o/p:  
Boy  
Cat  
Dog  
-------------------------------------------  
l=['Apple','Boy','Cat','Dog']  
for x in l:  
if len(x) != 3:  
print(x)  
o/p:  
Apple  
---------------------------------  
l=['Apple','Boy','Cat','Dog']  
for x in l:  
print(x)  
o/p:  
Apple  
Boy  
Cat  
Dog  
----------------------------------  
l=['Apple','Boy','Cat','Dog']  
l1=l[1:]  
for x in l1:  
print(x)  
o/p:  
Boy  
Cat  
Dog

Bottom of Form

Top of Form

Question 5: **Correct**

**Consider the following python code:**

1. **weight=62.4**
2. **zip='80098'**
3. **value=+23E4**

**The types of weight,zip and value variables respectively:**

* ​

int, str, float

* ​

float, str, str

* ​

float, str, float

**(Correct)**

* ​

double, str, float

**Explanation**

weight=62.4 is of float type, zip='80098' is of str type, value=+23E4 is of float type

Bottom of Form

Top of Form

Question 6: **Correct**

**You are writing a Python program. You required to handle data types properly. Consider the code segment:**

1. **a=10+20**
2. **b='10'+'20'**
3. **c='10'\*3**

**Identify the types of a,b and c?**

* ​

a is of int type,b is of int type and c is of int type

* ​

a is of int type,b is of str type and c is of str type

**(Correct)**

* ​

a is of int type ,b and c are invalid declarations

* ​

a is of int type,b is of str type and c is of int type

**Explanation**

If we apply + operator between two int types the result is of int type.  
a=10+20  
Hence a is int type.  
If we apply + operator between two str types, then the result is of str type.  
b='10'+'20'  
Hence b is str type  
c='10'\*3  
If we apply \* operator for 'str' and int type then it acts as string repetition operator. Hence the result is of str type

Bottom of Form

Top of Form

Question 7: **Correct**

**Consider the code:**

1. **s='Python is easy'**
2. **s1=s[-7:]**
3. **s2=s[-4:]**
4. **print(s1+s2)**

**What is the result?**

* ​

iseasyeasy

* ​

s easyeasy

* ​

is easyeasy

**(Correct)**

* ​

easyeasy

* ​

is easy easy

**Explanation**

s1 = s[-7:]===>'is easy'  
s2 = s[-4:]===>'easy'  
print(s1+s2)===>'is easyeasy'

Bottom of Form

Top of Form

Question 8: **Correct**

**Consider the code:**

**x= 8**

**y= 10**

**result= x//3\*3/2+y%2\*\*2**

**print(result)**

**What is the result?**

* ​

7.0

* ​

5.0

**(Correct)**

* ​

6.0

* ​

5

**Explanation**

x//3\*3/2+y%2\*\*2  
x//3\*3/2+y%4  
8//3\*3/2+y%4  
2\*3/2+y%4  
6/2+y%4  
3.0+10%4  
3.0+2  
5.0

Bottom of Form

Top of Form

Question 9: **Correct**

**You have the following code:**

**a=3**

**b=5**

**a += 2\*\*3**

**a -=b//2//3**

**print(a)**

**What is the result?**

* ​

10

* ​

12

* ​

13

* ​

11

**(Correct)**

**Explanation**

a+= 2\*\*3  
a = (a)+(2\*\*3) =a+8=3+8=11  
a -=b//2//3  
a = (a) - (b//2//3) =(a) - (5//2//3) =a-(2//3)=a-0=11

Bottom of Form

Top of Form

Question 10: **Incorrect**

**Consider the python code**

1. **a=1**
2. **b=3**
3. **c=5**
4. **d=7**

**In Which of the following cases the result value is 0?**

* ​

result = a%b-1

**(Correct)**

* ​

result = a+b\*2

* ​

result = a-b//d

**(Incorrect)**

* ​

result = a\*\*d-1

**(Correct)**

**Explanation**

a+b\*2--->7  
a%b-1-->0  
a-b//d--->1  
a\*\*d-1--->0

Bottom of Form

Top of Form

Question 11: **Correct**

**Consider the code**

1. **a=1**
2. **b=2**
3. **c=4**
4. **d=6**

**Which of the following expression results -4?**

* ​

(a+b)//d-c

**(Correct)**

* ​

(a+b)//c%d

* ​

(a+b)//c\*d

* ​

(b+c)//a%d

**Explanation**

(a+b)//c%d--->0  
(b+c)//a%d--->0  
(a+b)//c\*d---->0  
(a+b)//d-c---->-4

Bottom of Form

Top of Form

Question 12: **Correct**

**Consider the list:**

**list=['Apple','Banana','Carrot','Mango']**

**Which of the following are valid ways of accessing 'Mango':**

* ​

list[-1]

**(Correct)**

* ​

list[3]

**(Correct)**

* ​

list[4]

* ​

list[0]

**Explanation**

Python supports both positive and negative index.  
Positive index is from left to right and range is : 0 to length-1.  
Negative index is from right to left and range is : -1 to -length.  
In the above example we can access 'Mango' by using list[3] or list[-1]

Bottom of Form

Top of Form

Question 13: **Correct**

**Consider the code**

1. **a=2**
2. **a += 1**
3. **# Line-1**

**To make a value as 9,which expression required to place at Line-1**

* ​

a-=2

* ​

a+=2

* ​

a\*=2

* ​

a\*\*=2

**(Correct)**

**Explanation**

a\*=2--->6  
a\*\*=2---->9  
a+=2---->5  
a-=2--->1

Bottom of Form

Top of Form

Question 14: **Correct**

**Consider the following expression**

**result=(2\*(3+4)\*\*2-(3\*\*3)\*3)**

**What is result value?**

* ​

17

**(Correct)**

* ​

19

* ​

18

* ​

16

**Explanation**

Python Virtual Machine will give the precedence in the following order  
1. Parenthesis  
2. Exponent  
3. Multiplication,Division,Modulo,Floor Division  
4. Addition,Subtraction  
etc  
result=(2\*(3+4)\*\*2-(3\*\*3)\*3)=(2\*(7)\*\*2-(27)\*3)=2\*49-27\*3=98-81=17

Bottom of Form

Top of Form

Question 15: **Correct**

**Consider the following python code:**

1. **age=0**
2. **minor=False**
3. **name='Durga'**

**The types of age,minor and name variables respectively:**

* ​

float, bool, str

* ​

bool, bool, str

* ​

int, bool, char

* ​

int, bool, str

**(Correct)**

**Explanation**

age=0 is of int type, minor=False is of bool type, name='Durga' is of str type

Bottom of Form

Top of Form

Question 16: **Correct**

**Consider the lists:**

1. **numbers=[10,20,30,40,50]**
2. **alphabets=['a','b','c','d','e']**
3. **print( numbers is alphabets)**
4. **print( numbers == alphabets)**
5. **numbers=alphabets**
6. **print( numbers is alphabets)**
7. **print( numbers == alphabets)**

**What is the result?**

* ​

False

True

False

True

* ​

False

False

True

True

**(Correct)**

* ​

True

False

True

False

* ​

False

True

True

True

**Explanation**

'is' operator is always meant for reference comparison and == operator always meant for content comparison.

Bottom of Form

Top of Form

Question 17: **Correct**

**Consider the following lists:**

1. **n1=[10,20,30,40,50]**
2. **n2=[10,20,30,40,50]**
3. **print(n1 is n2)**
4. **print(n1 == n2)**

**What is the output?**

* ​

True

True

* ​

False

True

**(Correct)**

* ​

True

False

* ​

False

False

**Explanation**

'is' operator is always meant for reference comparison and == operator always meant for content comparison.  
In the above example n1 and n2 are different objects but with same content.  
Hence 'is' operator returns False and '==' operator returns True

Bottom of Form

Top of Form

Question 18: **Correct**

**You are developing a python application for your company.**

**A list named employees contains 500 employee names.**

**In which cases we will get IndexError while accessing employee names?**

* ​

None of the above

* ​

employees[0]

* ​

employees[-1]

* ​

employees[500]

**(Correct)**

**Explanation**

If we are trying to access list elements with out of range index, then we will get IndexError.  
In the above example, list contains 500 names and hence valid positive index range is 0 to 499 and negative index range is -1 to -500.  
employees[500]=====>IndexError

Bottom of Form

Top of Form

Question 19: **Correct**

**Consider the code**

1. **s='AB CD'**
2. **list=list(s)**
3. **list.append('EF')**
4. **print(list)**

**What is the result?**

* ​

['A','B','C','D','EF']

* ​

{'A','B',' ','C','D','EF'}

* ​

['A','B','C','D','E','F']

* ​

('A','B',' ','C','D','EF')

* ​

['A','B',' ','C','D','EF']

**(Correct)**

**Explanation**

List elements will be printed within square brackets.  
Whenever we are converting string to list,each character will become element of List including space also.

Bottom of Form

Top of Form

Question 20: **Incorrect**

**Consider the following code**

1. **x= 'Durga'**
2. **y= 'Durga'**
3. **result=condition**
4. **print(result)**

**For which of the following condition True will be printed to the console?**

* ​

x != y

* ​

x < y

* ​

x is y

**(Correct)**

* ​

x is not y

**(Incorrect)**

**Explanation**

Both x and y pointing to the same object.  
Hence 'x is y' returns True. Except that all remaining cases returns False.  
x is y==>True  
x is not y===>False  
x != y==>False  
x < y===>False

Bottom of Form

Top of Form

Question 21: **Correct**

1. **subjects=['java','python','sap']**
2. **more\_subjects=['java','python','sap']**
3. **extra\_subjects=more\_subjects**

**In which cases True will be printed to the console?**

* ​

print(subjects is extra\_subjects)

* ​

print(extra\_subjects is more\_subjects)

**(Correct)**

* ​

print(subjects is more\_subjects)

* ​

print(subjects == extra\_subjects)

**(Correct)**

**Explanation**

We can use 'is' operator for reference comparison where as == operator for content comparison.  
print(extra\_subjects is more\_subjects) #True  
print(subjects is more\_subjects) #False  
print(subjects is extra\_subjects)#False  
print(subjects == extra\_subjects)#True

Bottom of Form

Top of Form

Question 22: **Incorrect**

**Consider the python code**

1. **numbers=[10,20,30,40]**
2. **x=0**

**In which of the following cases 10 will be printed to the console?**

* ​
  1. for i in (30,40,50):
  2. if i in numbers:
  3. x=x+5
  4. print(x)

**(Correct)**

* ​
  1. for i in (30,40,50):
  2. if i not in numbers:
  3. x=x+10
  4. print(x)

**(Correct)**

* ​
  1. for i in (30,40,50):
  2. if i not in numbers:
  3. x=x+5
  4. print(x)

**(Incorrect)**

* ​
  1. for i in (30,40,50):
  2. if i in numbers:
  3. x=x+10
  4. print(x)

**Explanation**

for i in (30,40,50):  
if i in numbers:  
x=x+5  
print(x) #10  
----------------------------  
for i in (30,40,50):  
if i not in numbers:  
x=x+5  
print(x)#5  
----------------------  
for i in (30,40,50):  
if i not in numbers:  
x=x+10  
print(x)#10  
--------------------------  
for i in (30,40,50):  
if i in numbers:  
x=x+10  
print(x) #20

Bottom of Form

Top of Form

Question 23: **Incorrect**

**Consider the code:**

**a=21**

**b=6**

**print(a/b)**

**print(a//b)**

**print(a%b)**

**What is the result?**

* ​

3

3

3

* ​

3.5

3.5

3

* ​

3.0

3

3

**(Incorrect)**

* ​

3.5

3

3

**(Correct)**

**Explanation**

division operator in python always meant for floating point arithmetic. Hence a/b returns 3.5 But floor division(//) operator can perform both integral and floating point arithmetic. If the arguments are int type then the result is int type and if the arguments are float type then the result is float type. Hence a//b returns 3. a%b returns the remainder which is 3.

Bottom of Form

Top of Form

Question 24: **Correct**

**Consider the code:**

1. **x='ACROTE'**
2. **y='APPLE'**
3. **z='TOMATO'**

**Which of the following won't print 'CAT' to the console**

* ​

print(x[-5]+y[0]+z[0])

* ​

print(x[-5]+y[0]+z[-2])

* ​

print(x[1]+y[0]+z[0])

* ​

print(x[2]+y[1]+z[1])

**(Correct)**

**Explanation**

print(x[1]+y[0]+z[0]) #CAT  
print(x[2]+y[1]+z[1]) #RPO  
print(x[-5]+y[0]+z[0]) #CAT  
print(x[-5]+y[0]+z[-2]) #CAT

Bottom of Form

Top of Form

Question 25: **Incorrect**

**You have the following code:**

1. **a=bool([False])**
2. **b=bool(3)**
3. **c=bool("")**
4. **d=bool(' ')**

**Which of the variables will represent False:**

* ​

c

**(Correct)**

* ​

a

* ​

d

* ​

b

**(Incorrect)**

**Explanation**

For Empty String, Empty List,Empty tuple,Empty set,Empty dict and range(0) arguments bool() function returns False.  
c=bool("")  
As the argument is empty string, it represents False.

Bottom of Form

Top of Form

Question 26: **Correct**

**Consider the following code segments:**

**# Code Segment-1**

1. **a1='10'**
2. **b1=3**
3. **c1=a1\*b1**

**# Code Segment-2**

1. **a2=10**
2. **b2=3**
3. **c2=a2/b2**

**# Code Segment-3**

1. **a3=2.6**
2. **b3=1**
3. **c3=a3/b3**

**After executing Code Segments 1,2 and 3 the result types of c1,c2 and c3 are:**

* ​

c1 is of str type,c2 is of float type ,c3 is of float type

**(Correct)**

* ​

c1 is of str type,c2 is of int type ,c3 is of float type

* ​

c1 is of str type,c2 is of int type ,c3 is of int type

* ​

c1 is of str type,c2 is of str type ,c3 is of str type

**Explanation**

The value of c1 is '101010', which is str type  
The value of c2 is 3.3333, which is float type  
The value of c3 is 2.6,which is float type

Bottom of Form

Top of Form

Question 27: **Incorrect**

**Consider the code:**

1. **s='Python is easy'**
2. **s1=s[6:-4]**
3. **#Line-1**
4. **print(len(s2))**

**To print 2 as output,which code we have to insert at Line-1**

* ​

s2 = s1.rstrip()

* ​

s2 = s1.lstrip()

**(Incorrect)**

* ​

s2 = s1.lrstrip()

* ​

s2 = s1.strip()

**(Correct)**

**Explanation**

strip()==>It will remove spaces present at left and right sides of the string  
lstrip()==>It will remove spaces present at only left side of the string  
rstrip()==>It will remove spaces present at only right side of the string  
There is no method like lrstrip().  
s2 = s1.lrstrip()  
AttributeError: 'str' object has no attribute 'lrstrip'

Bottom of Form

Top of Form

Question 28: **Correct**

**Consider the following lists:**

1. **n1=[10,20,30,40,50]**
2. **n2=[10,20,30,40,50]**
3. **print(n1 is n2)**
4. **print(n1 == n2)**
5. **n1=n2**
6. **print(n1 is n2)**
7. **print(n1 == n2)**

**What is the result?**

* ​

False

False

True

True

* ​

True

False

True

False

* ​

False

True

False

True

* ​

False

True

True

True

**(Correct)**

**Explanation**

'is' operator is always meant for reference comparison and == operator always meant for content comparison.

Bottom of Form

Top of Form

Question 29: **Correct**

**You are writing a python program that evaluates an arithmetic expression.**

**The expression is described as b is equals a multiplied by negative one,then raised to the second power,where a is the value which will be input and b is result.**

**a=eval(input('Enter a number for the expression:'))**

**Which of the following is valid expression for the given requirement?**

* ​

b = (a-)\*\*2

* ​

b = (a)\*\*-2

* ​

b = -(a)\*\*2

* ​

b = (-a)\*\*2

**(Correct)**

**Explanation**

b = (-a)\*\*2  
b is equals a multiplied by negative one,then raised to the second power

Bottom of Form

Top of Form

Question 30: **Incorrect**

**You are developing a python application for your company.**

**A list named employees contains 600 employee names,the last 3 being company management.  You need to slice employees to display all employees excluding management. Which two code segments we should use?**

* ​

employees[0:-2]

**(Incorrect)**

* ​

employees[:-3]

**(Correct)**

* ​

employees[0:-3]

**(Correct)**

* ​

employees[1:-2]

* ​

employees[1:-3]

**Explanation**

list[begin:end] returns list of elements from begin index to end-1 index default value for begin is: 0

Bottom of Form

Top of Form

Question 31: **Correct**

**Consider the code**

1. **a=15**
2. **b=5**
3. **print(a/b)**

**What is the result ?**

* ​

3.0

**(Correct)**

* ​

0

* ​

0.0

* ​

3

**Explanation**

/ always meant for floating point arithmetic

Bottom of Form

Top of Form

Question 32: **Correct**

**You are writing a Python program to read two int values from the keyboard and print the sum.**

1. **x=input('Enter First Number:')**
2. **y=input('Enter Second Number:')**
3. **#Line-1**

**Which of the following code we have to write at Line-1 to print sum of given numbers?**

* ​

print('The Result:'+(int(x)+int(y)))

* ​

print('The Result:'+(int(x+y)))

* ​

print('The Result:'+str(int(x+y)))

* ​

print('The Result:'+str(int(x)+int(y)))

**(Correct)**

**Explanation**

To use + operator for string types, compulsory both arguments must be str type, otherwise we will get error.  
print('The Result:'+(int(x)+int(y))) #TypeError: must be str, not int  
print('The Result:'+(int(x+y))) #TypeError: unsupported operand type(s) for +: 'str' and 'str'  
print('The Result:'+str(int(x)+int(y))) #valid  
print('The Result:'+str(int(x+y)))# won't fulfill our requirement

Bottom of Form

Top of Form

Question 33: **Correct**

**Which of the following is valid python operator precedence order?**

* ​

Exponents

Parenthesis

Unary Positive,Negative and Not

Multiplication and Division

Addition and Subtraction

And

* ​

Exponents

Unary Positive,Negative and Not

Multiplication and Division

Addition and Subtraction

And

Parenthesis

* ​

Parenthesis

Exponents

Unary Positive,Negative and Not

Addition and Subtraction

Multiplication and Division

And

* ​

Parenthesis

Exponents

Unary Positive,Negative and Not

Multiplication and Division

Addition and Subtraction

And

**(Correct)**

**Explanation**

The following is the correct order of Python Operator Precedence  
Parenthesis  
Exponents  
Unary Positive,Negative and Not  
Multiplication and Division  
Addition and Subtraction  
And

Bottom of Form

Top of Form

Question 34: **Correct**

**Consider the Python code:**

1. **a=5**
2. **b=10**
3. **c=2**
4. **d=True**
6. **x=a+b\*c**
7. **y=a+b/d**
9. **if(condition):**
10. **print('Valid')**
11. **else:**
12. **print('invalid')**

**To print 'Valid' to the console, which condition we have to take for if statement?**

* ​

x<y

* ​

x>y

**(Correct)**

* ​

x<=y

* ​

x==y

**Explanation**

a=5  
b=10  
c=2  
d=True  
x=a+b\*c=5+10\*2=5+20=25  
y=a+b/d=5+10/1=5+10.0=15.0  
To print valid the condition should be True.It is possible if condition is x>y.  
xFalse  
x<=y===>False  
x>y===>True  
x==y===>False

Bottom of Form

Top of Form

Question 35: **Correct**

**Consider the following variable declarations:**

1. **a= bool([])**
2. **b= bool(())**
3. **c= bool(range(0))**
4. **d= bool({})**
5. **e= bool(set())**

**Which of the above variables represent True ?**

* ​

c

* ​

All Variables represent True

* ​

a ,b, c, d

* ​

None of the variables represents True

**(Correct)**

**Explanation**

For Empty String, Empty List,Empty tuple,Empty set,Empty dict and range(0) arguments bool() function returns False.

Bottom of Form

Top of Form

Question 36: **Correct**

**Consider the code:**

1. **start=input('How old were you at the time of joining?')**
2. **end=input('How old are you today?')**

**Which of the following code is valid to print Congratulations message?**

* ​

print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')

* ​

print('Congratulations on '+ str(end-start)+' Years of Service!')

* ​

print('Congratulations on '+ int(end-start)+' Years of Service!')

* ​

print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')

**(Correct)**

**Explanation**

To use + operator for string types, compulsory both arguments must be str type, otherwise we will get error.  
print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')  
TypeError: must be str, not int  
print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')  
print('Congratulations on '+ int(end-start)+' Years of Service!')  
TypeError: unsupported operand type(s) for -: 'str' and 'str'  
print('Congratulations on '+ str(end-start)+' Years of Service!')  
TypeError: unsu'str'TypeError: unsupported operand type(s) for -: 'str' and 'str'pported operand type(s) for -: 'str' and

Bottom of Form

Top of Form

Question 37: **Correct**

**Consider the expession:**

**result=a-b\*c+d**

**Which of the following are valid?**

* ​

The above expession is equivalent to a-(b\*c)+d

**(Correct)**

* ​

First a-b will be evaluated followed by multiplication and addition

* ​

First b\*c will be evaluated followed by subtraction and addition

**(Correct)**

* ​

First b\*c will be evaluated followed by addition and subtraction

**Explanation**

multiplication having more precedence than addition and subtraction. addition and subtraction having same precedence.

Bottom of Form

Top of Form

Question 38: **Incorrect**

1. **a=bool(0)**
2. **b=bool(3)**
3. **c=bool(0.5)**
4. **d=bool(0.0)**

**Which variables represent True?**

* ​

d,a

* ​

b,c

**(Correct)**

* ​

a,b

* ​

c,d

* ​

All Variables

**(Incorrect)**

**Explanation**

In the case of integral values 0 treated as False and non-zero treated as True. In the case of float values 0.0 treated as False and all other values (non-zero values) treated as True

Bottom of Form

Top of Form

Question 39: **Incorrect**

**Consider the following expression**

**result=8//6%5+2\*\*3-2**

**print(result)**

**What is the result?**

* ​

8

* ​

9

* ​

7

**(Correct)**

* ​

6

**(Incorrect)**

**Explanation**

2\*\*3=8  
8//6=1  
result=1%5+8-2=1+8-2=9-2=7

Bottom of Form

Top of Form

Question 40: **Incorrect**

**You are developing a python application for your company.**

**A list named employees contains 500 employee names.**

**In which cases we will get IndexError while accessing employee data?**

* ​

employees[0:501]

* ​

None of the above

**(Correct)**

* ​

employees[-10:10]

* ​

employees[1:1000]

**(Incorrect)**

**Explanation**

Slice Operator never raises IndexError

Bottom of Form