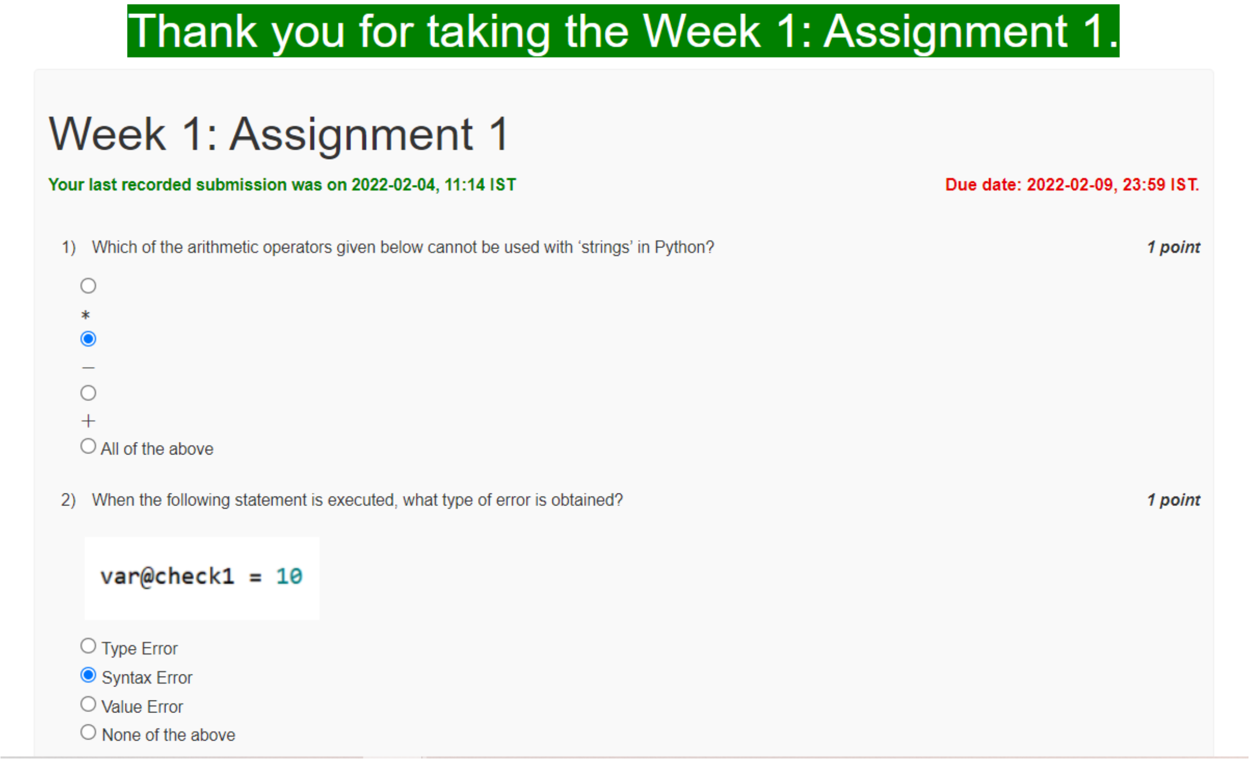


Thank you for taking the Week 1: Assignment 1.

Week 1: Assignment 1

Your last recorded submission was on 2022-02-04, 11:14 IST

Due date: 2022-02-09, 23:59 IST.

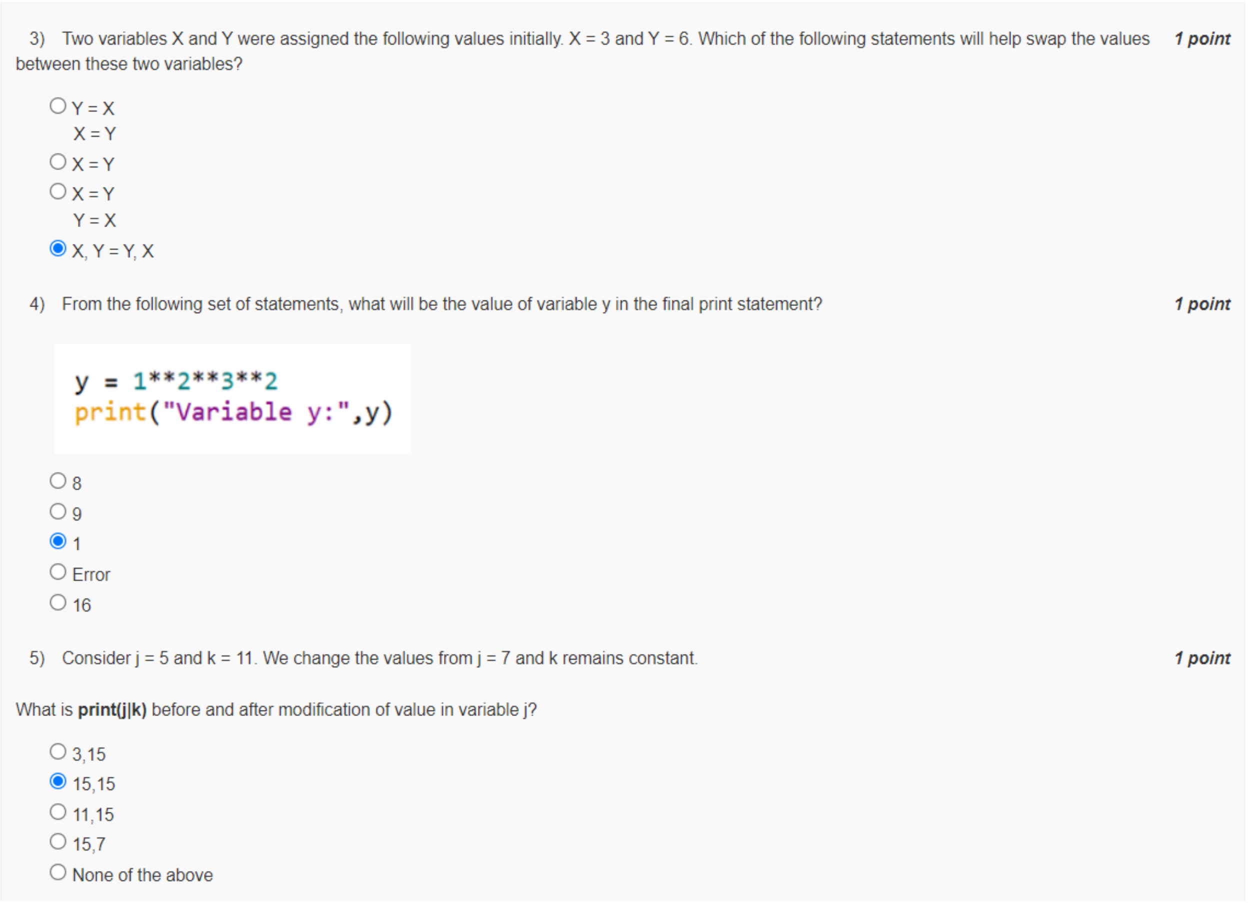
1) Which of the arithmetic operators given below cannot be used with 'strings' in Python? *1 point*

○
*
●
–
○
+
○ All of the above

2) When the following statement is executed, what type of error is obtained? *1 point*

```
var@check1 = 10
```

○ Type Error
● Syntax Error
○ Value Error
○ None of the above



3) Two variables X and Y were assigned the following values initially. X = 3 and Y = 6. Which of the following statements will help swap the values between these two variables? *1 point*

○ Y = X
  X = Y
○ X = Y
○ X = Y
  Y = X
● X, Y = Y, X

4) From the following set of statements, what will be the value of variable y in the final print statement? *1 point*

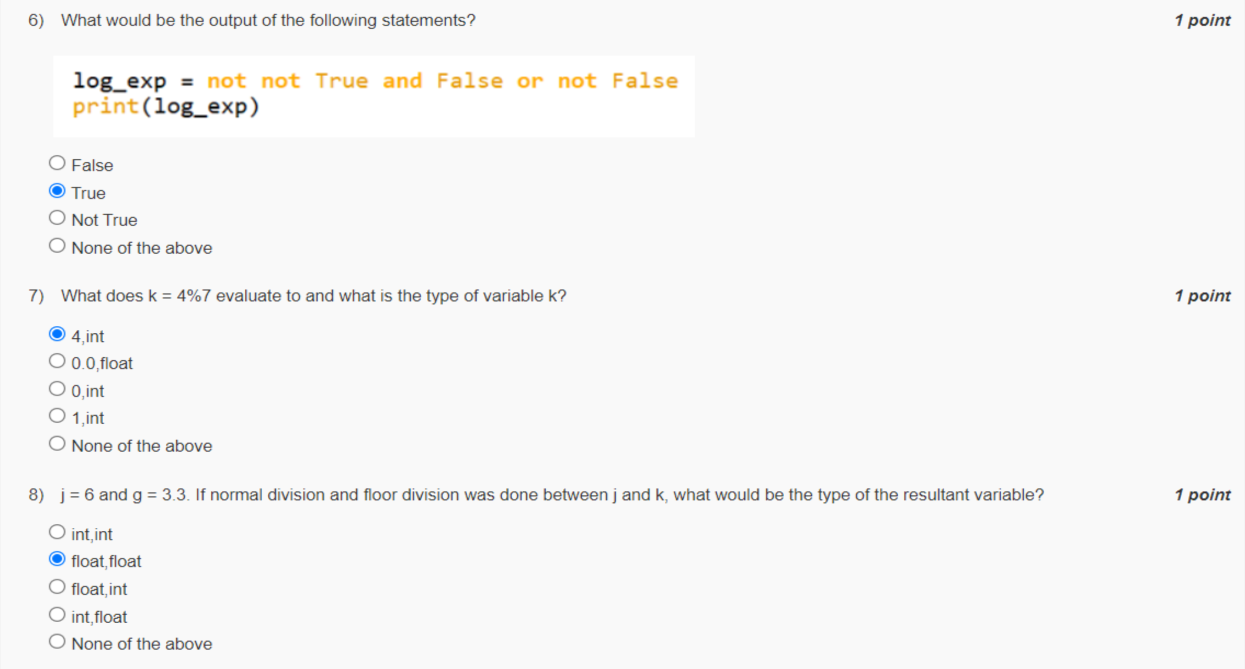
```
y = 1**2**3**2
print("Variable y:",y)
```

○ 8
○ 9
● 1
○ Error
○ 16

5) Consider j = 5 and k = 11. We change the values from j = 7 and k remains constant. *1 point*

What is **print(j|k)** before and after modification of value in variable j?

○ 3,15
● 15,15
○ 11,15
○ 15,7
○ None of the above



6) What would be the output of the following statements? *1 point*

```
log_exp = not not True and False or not False
print(log_exp)
```

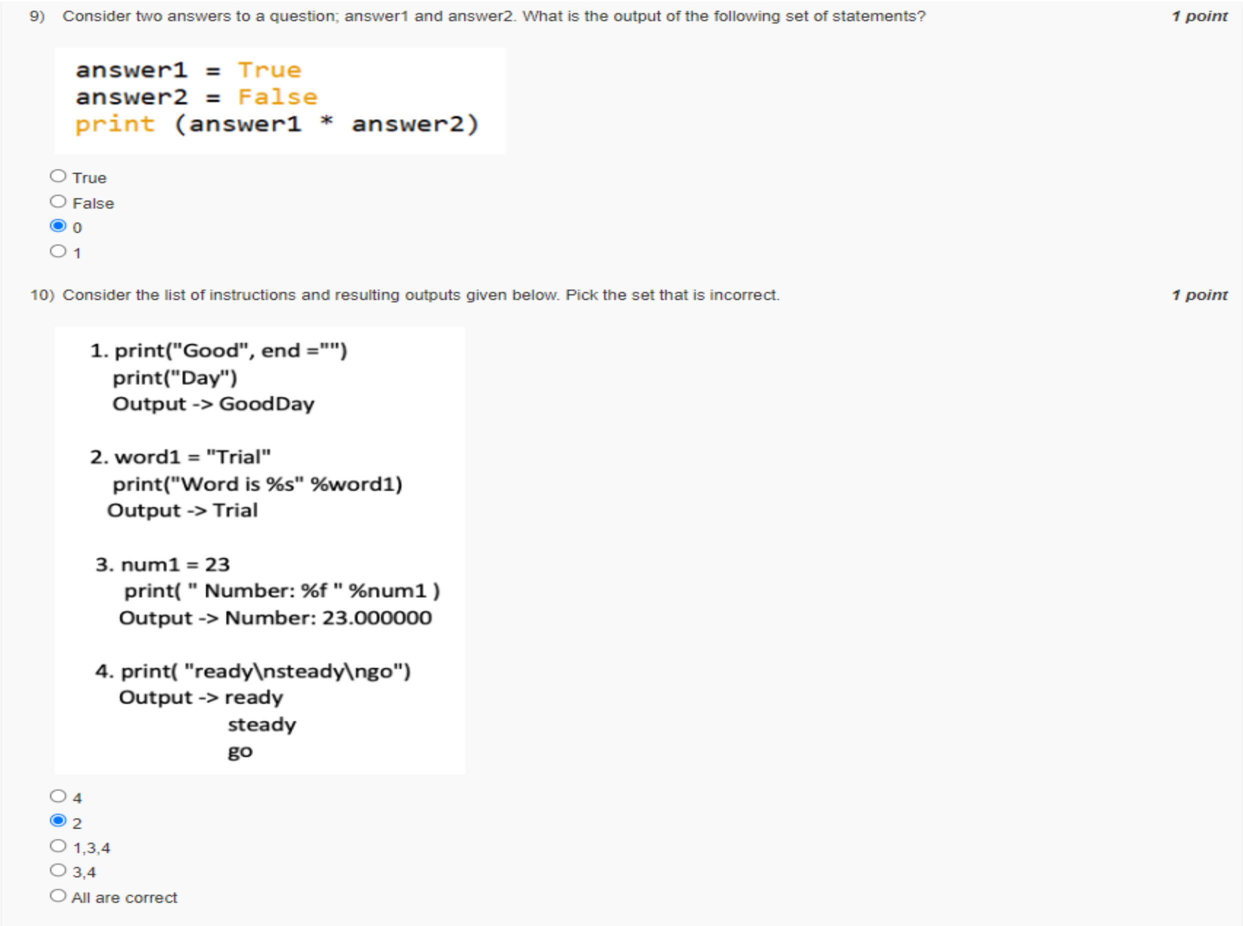
○ False
● True
○ Not True
○ None of the above

7) What does k = 4%7 evaluate to and what is the type of variable k? *1 point*

● 4,int
○ 0.0,float
○ 0,int
○ 1,int
○ None of the above

8) j = 6 and g = 3.3. If normal division and floor division was done between j and k, what would be the type of the resultant variable? *1 point*

○ int,int
● float,float
○ float,int
○ int,float
○ None of the above



9) Consider two answers to a question; answer1 and answer2. What is the output of the following set of statements? *1 point*

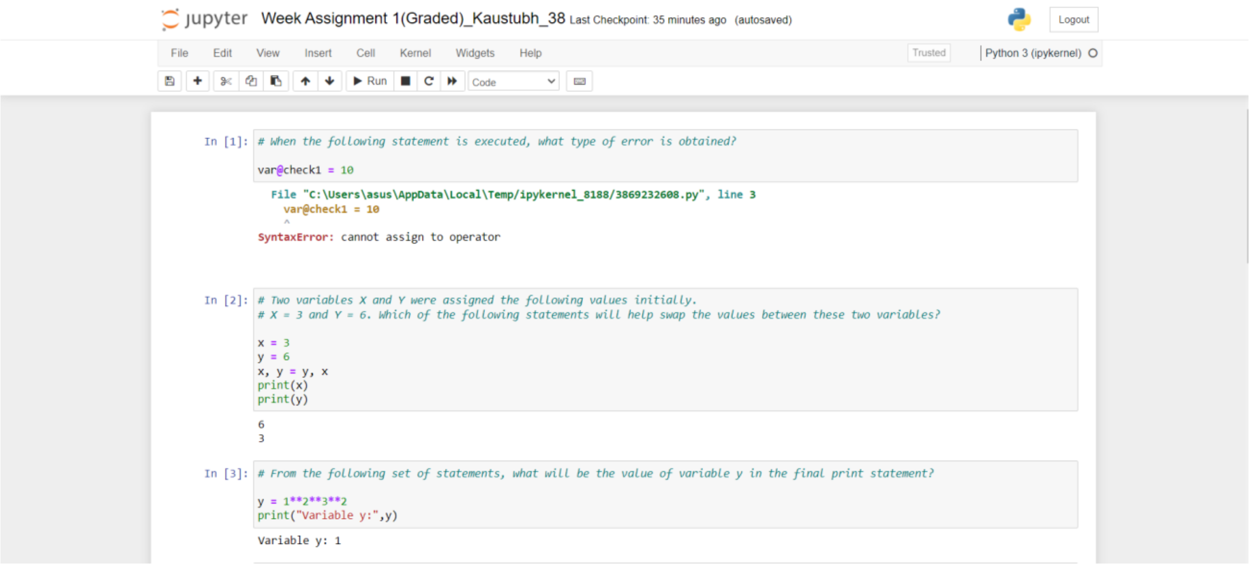
```
answer1 = True
answer2 = False
print (answer1 * answer2)
```

○ True
○ False
● 0
○ 1

10) Consider the list of instructions and resulting outputs given below. Pick the set that is incorrect. *1 point*

```
1. print("Good", end ="")
   print("Day")
   Output -> GoodDay

2. word1 = "Trial"
   print("Word is %s" %word1)
  Output -> Trial

 3. num1 = 23
     print( " Number: %f " %num1 )
    Output -> Number: 23.000000

 4. print( "ready\nsteady\ngo")
    Output -> ready
                    steady
                    go
```

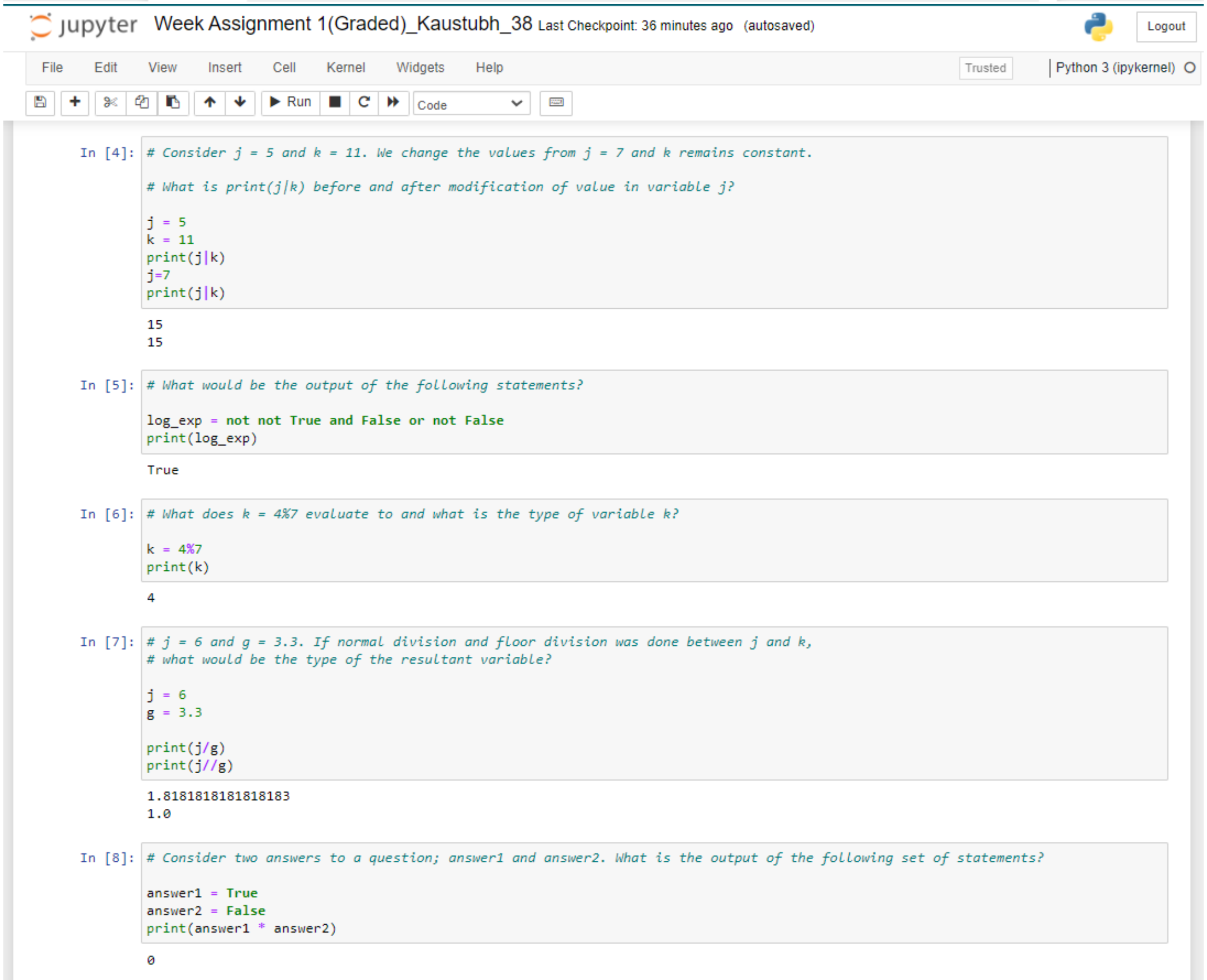
○ 4
● 2
○ 1,3,4
○ 3,4
○ All are correct



```
In [1]: # When the following statement is executed, what type of error is obtained?

        var@check1 = 10

          File "C:\Users\asus\AppData\Local\Temp/ipykernel_8188/3869232608.py", line 3
            var@check1 = 10
            ^
        SyntaxError: cannot assign to operator
```

```
In [2]: # Two variables X and Y were assigned the following values initially.
        # X = 3 and Y = 6. Which of the following statements will help swap the values between these two variables?

        x = 3
        y = 6
        x, y = y, x
        print(x)
        print(y)

        6
        3
```

```
In [3]: # From the following set of statements, what will be the value of variable y in the final print statement?

        y = 1**2**3**2
        print("Variable y:",y)

        Variable y: 1
```



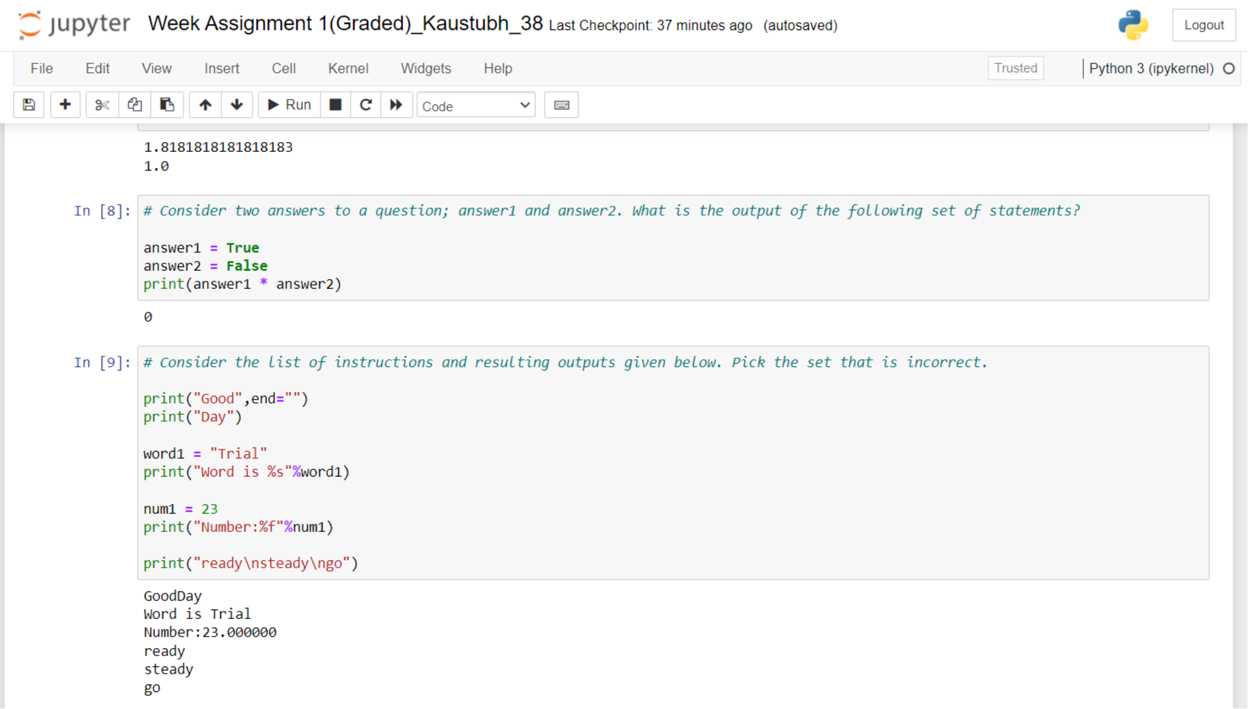
```
In [4]: # Consider j = 5 and k = 11. We change the values from j = 7 and k remains constant.

        # What is print(j|k) before and after modification of value in variable j?

        j = 5
        k = 11
        print(j|k)
        j=7
        print(j|k)

        15
        15
```

```
In [5]: # What would be the output of the following statements?

        log_exp = not not True and False or not False
        print(log_exp)

        True
```

```
In [6]: # What does k = 4%7 evaluate to and what is the type of variable k?

        k = 4%7
        print(k)

        4
```

```
In [7]: # j = 6 and g = 3.3. If normal division and floor division was done between j and k,
        # what would be the type of the resultant variable?

        j = 6
        g = 3.3

        print(j/g)
        print(j//g)

        1.8181818181818183
        1.0
```

```
In [8]: # Consider two answers to a question; answer1 and answer2. What is the output of the following set of statements?

        answer1 = True
        answer2 = False
        print(answer1 * answer2)

        0
```



```
1.8181818181818183
1.0
```

In [8]:

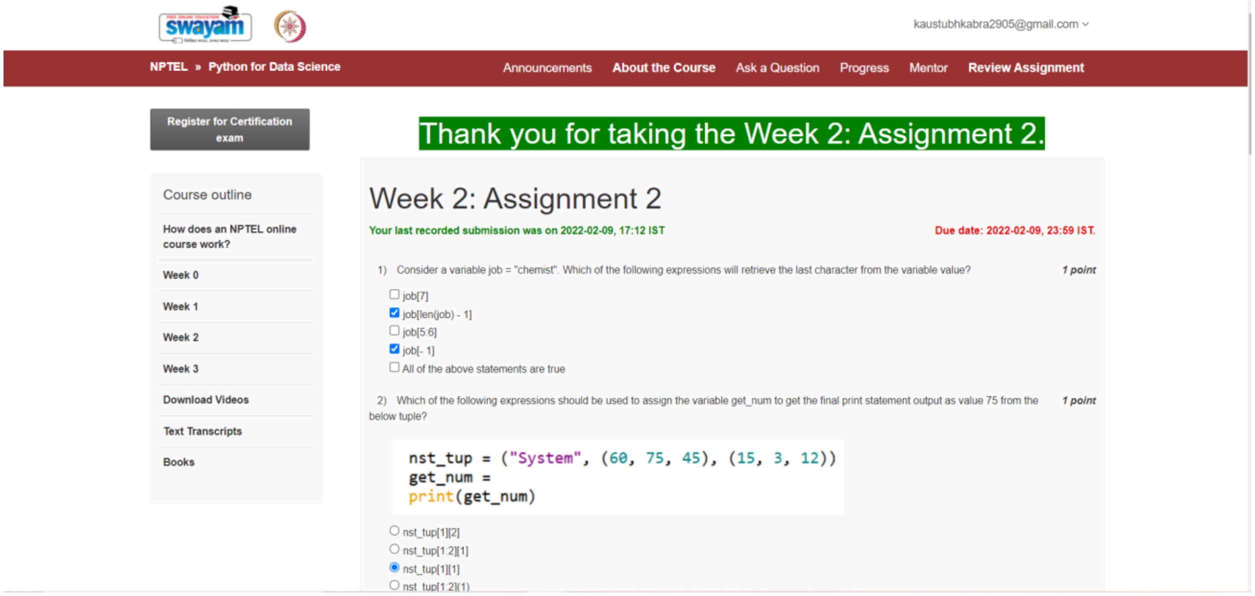
```python
# Consider two answers to a question; answer1 and answer2. What is the output of the following set of statements?

answer1 = True
answer2 = False
print(answer1 * answer2)
```

```
0
```

In [9]:

```python
# Consider the list of instructions and resulting outputs given below. Pick the set that is incorrect.

print("Good",end="")
print("Day")

word1 = "Trial"
print("Word is %s"%word1)

num1 = 23
print("Number:%f"%num1)

print("ready\nsteady\ngo")
```

```
GoodDay
Word is Trial
Number:23.000000
ready
steady
go
```



Thank you for taking the Week 2: Assignment 2.

Week 2: Assignment 2

Your last recorded submission was on 2022-02-09, 17:12 IST

Due date: 2022-02-09, 23:59 IST.

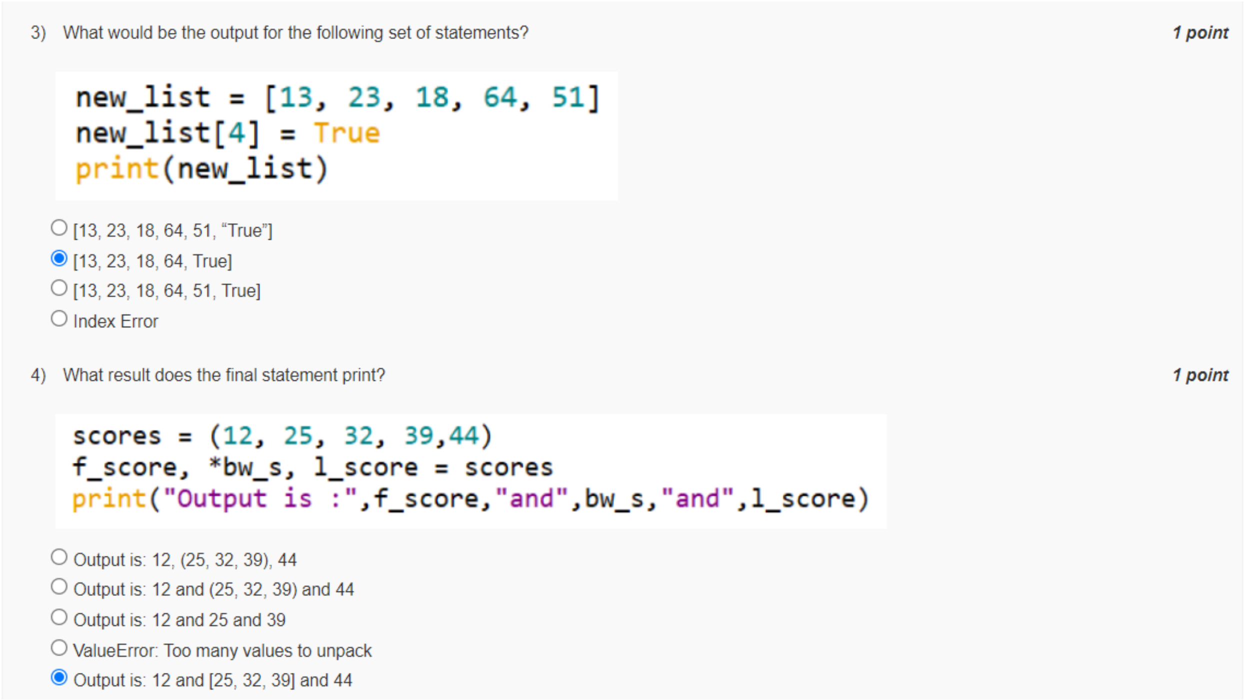
1) Consider a variable job = "chemist". Which of the following expressions will retrieve the last character from the variable value? *1 point*

☐ job[7]
☑ job[len(job) - 1]
☐ job[5:6]
☑ job[- 1]
☐ All of the above statements are true

2) Which of the following expressions should be used to assign the variable get_num to get the final print statement output as value 75 from the below tuple? *1 point*

```
nst_tup = ("System", (60, 75, 45), (15, 3, 12))
get_num =
print(get_num)
```

○ nst_tup[1][2]
○ nst_tup[1:2][1]
● nst_tup[1][1]
○ nst_tup[1:2](1)



3) What would be the output for the following set of statements? *1 point*

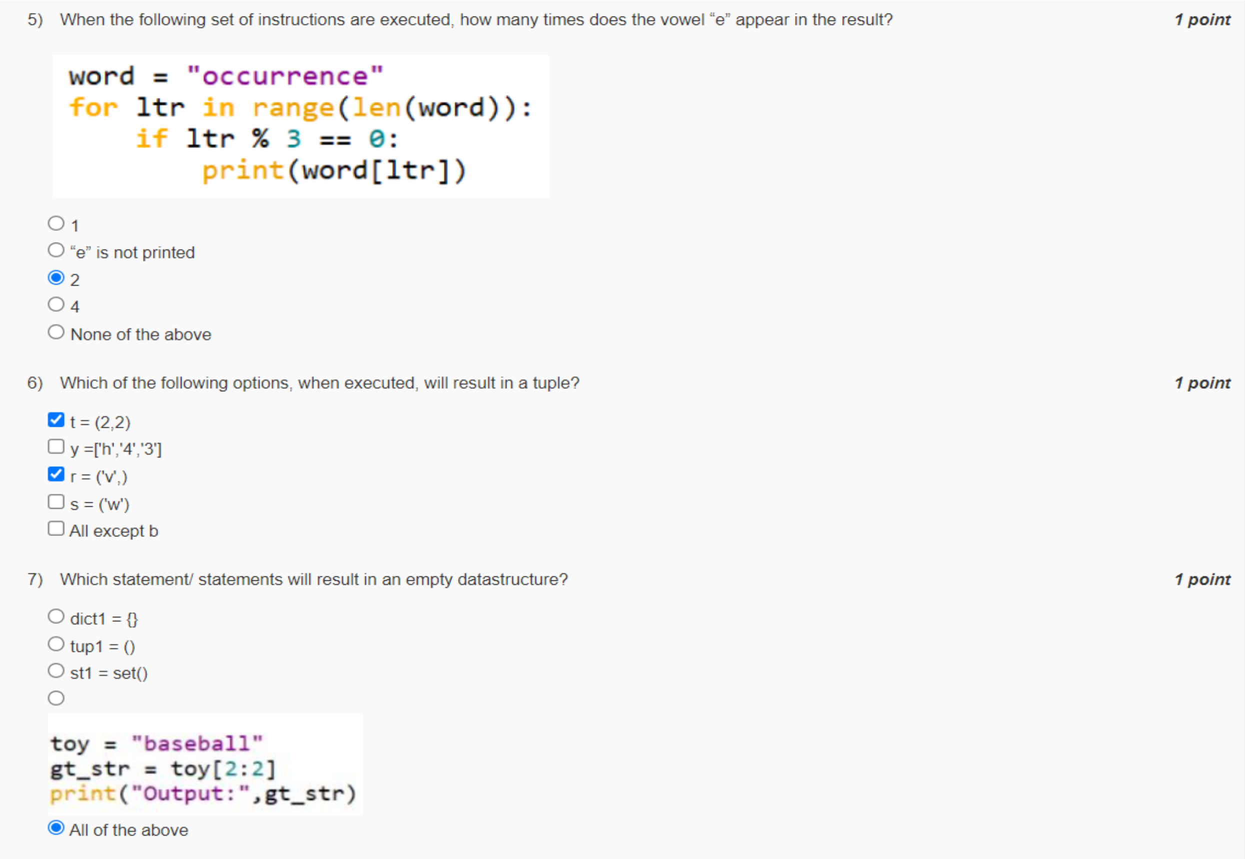
```
new_list = [13, 23, 18, 64, 51]
new_list[4] = True
print(new_list)
```

○ [13, 23, 18, 64, 51, "True"]
● [13, 23, 18, 64, True]
○ [13, 23, 18, 64, 51, True]
○ Index Error

4) What result does the final statement print? *1 point*

```
scores = (12, 25, 32, 39,44)
f_score, *bw_s, l_score = scores
print("Output is :",f_score,"and",bw_s,"and",l_score)
```

○ Output is: 12, (25, 32, 39), 44
○ Output is: 12 and (25, 32, 39) and 44
○ Output is: 12 and 25 and 39
○ ValueError: Too many values to unpack
● Output is: 12 and [25, 32, 39] and 44



5) When the following set of instructions are executed, how many times does the vowel “e” appear in the result? *1 point*

```
word = "occurrence"
for ltr in range(len(word)):
    if ltr % 3 == 0:
        print(word[ltr])
```

○ 1
○ “e” is not printed
● 2
○ 4
○ None of the above

6) Which of the following options, when executed, will result in a tuple? *1 point*

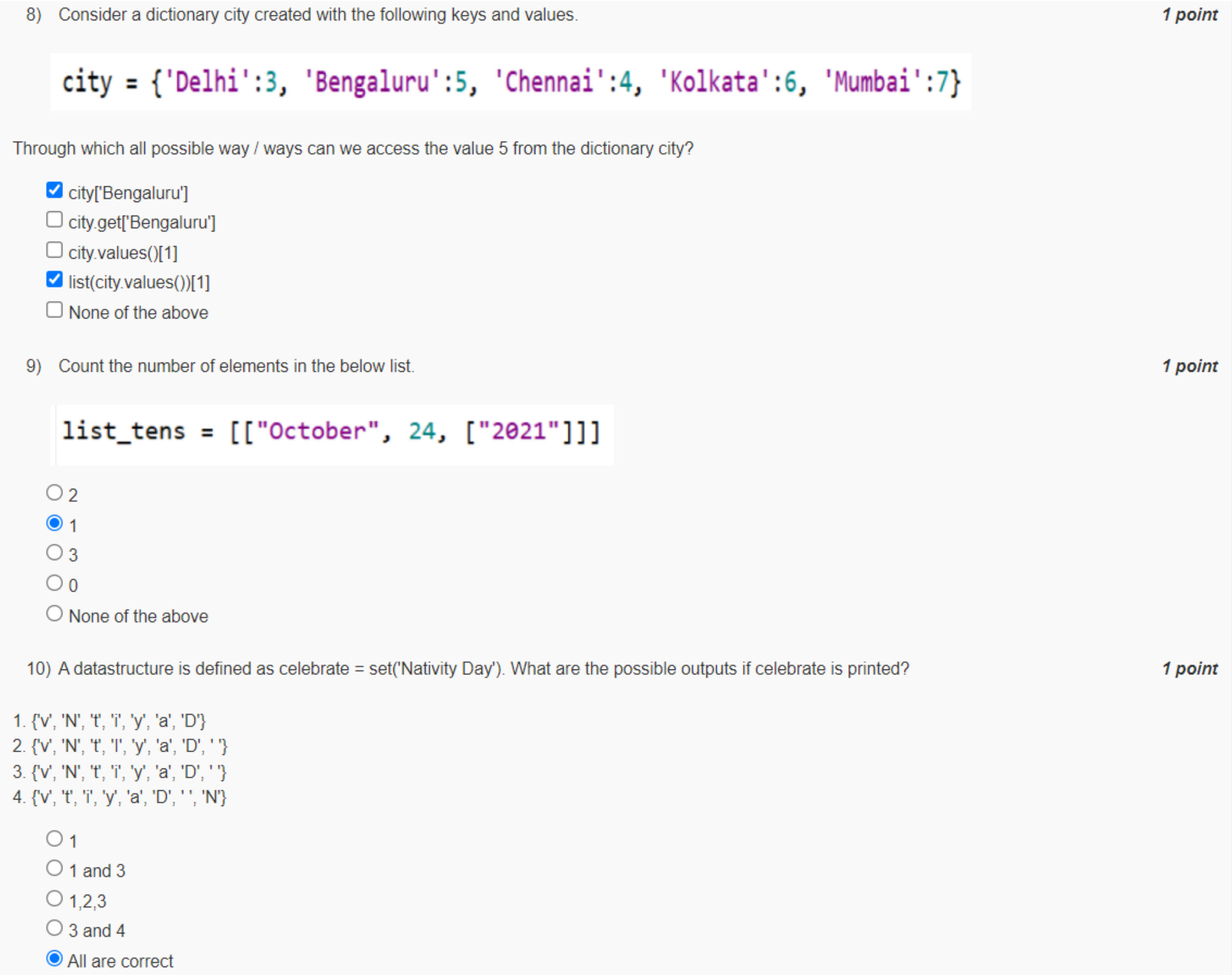
☑ t = (2,2)
□ y =['h','4','3']
☑ r = ('v',)
□ s = ('w')
□ All except b

7) Which statement/ statements will result in an empty datastructure? *1 point*

○ dict1 = {}
○ tup1 = ()
○ st1 = set()
○

```
toy = "baseball"
gt_str = toy[2:2]
print("Output:",gt_str)
```

● All of the above



8) Consider a dictionary city created with the following keys and values. *1 point*

```
city = {'Delhi':3, 'Bengaluru':5, 'Chennai':4, 'Kolkata':6, 'Mumbai':7}
```

Through which all possible way / ways can we access the value 5 from the dictionary city?

☑ city['Bengaluru']
□ city.get['Bengaluru']
□ city.values()[1]
☑ list(city.values())[1]
□ None of the above

9) Count the number of elements in the below list. *1 point*

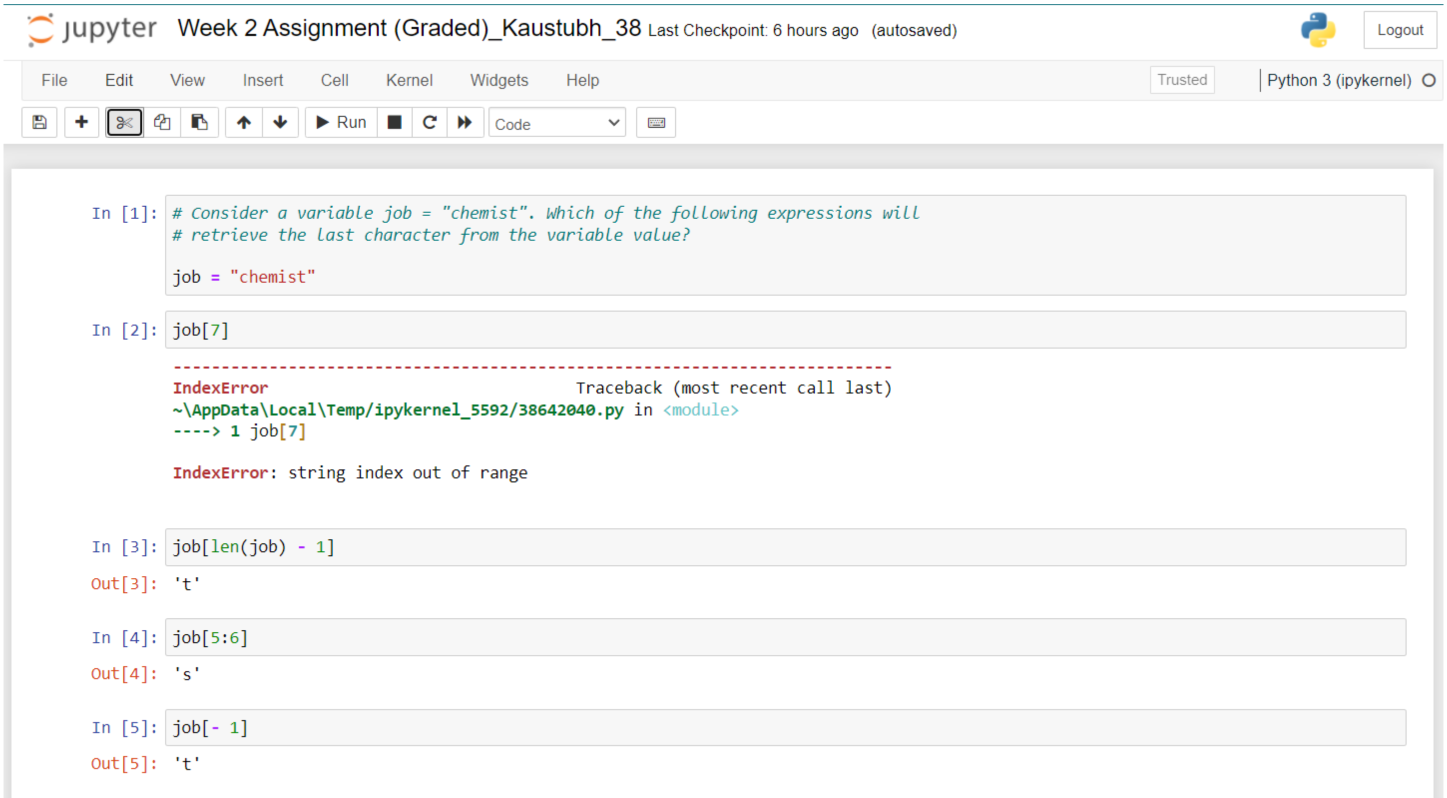
```
list_tens = [["October", 24, ["2021"]]]
```

○ 2
● 1
○ 3
○ 0
○ None of the above

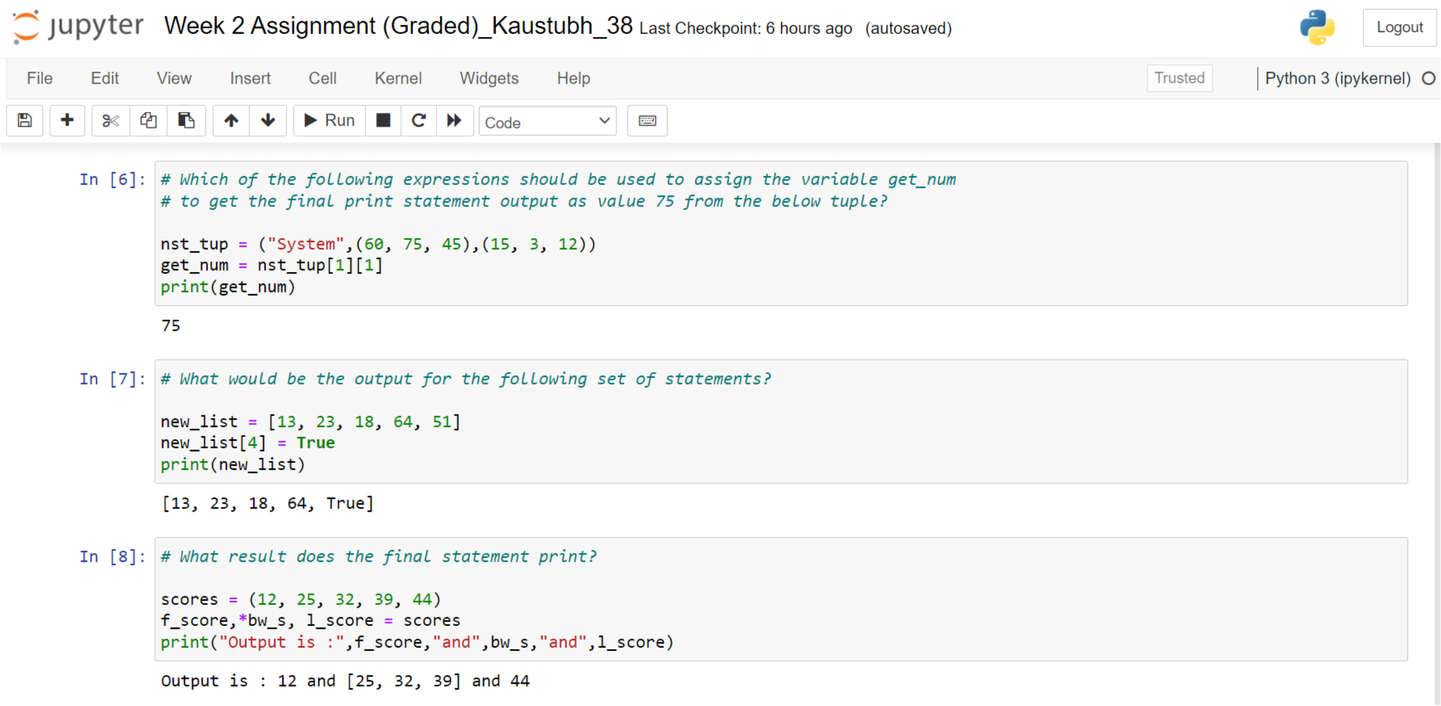
10) A datastructure is defined as celebrate = set('Nativity Day'). What are the possible outputs if celebrate is printed? *1 point*

1. {'v', 'N', 't', 'i', 'y', 'a', 'D'}
2. {'v', 'N', 't', 'I', 'y', 'a', 'D', ' '}
3. {'v', 'N', 't', 'i', 'y', 'a', 'D', ' '}
4. {'v', 't', 'i', 'y', 'a', 'D', ' ', 'N'}

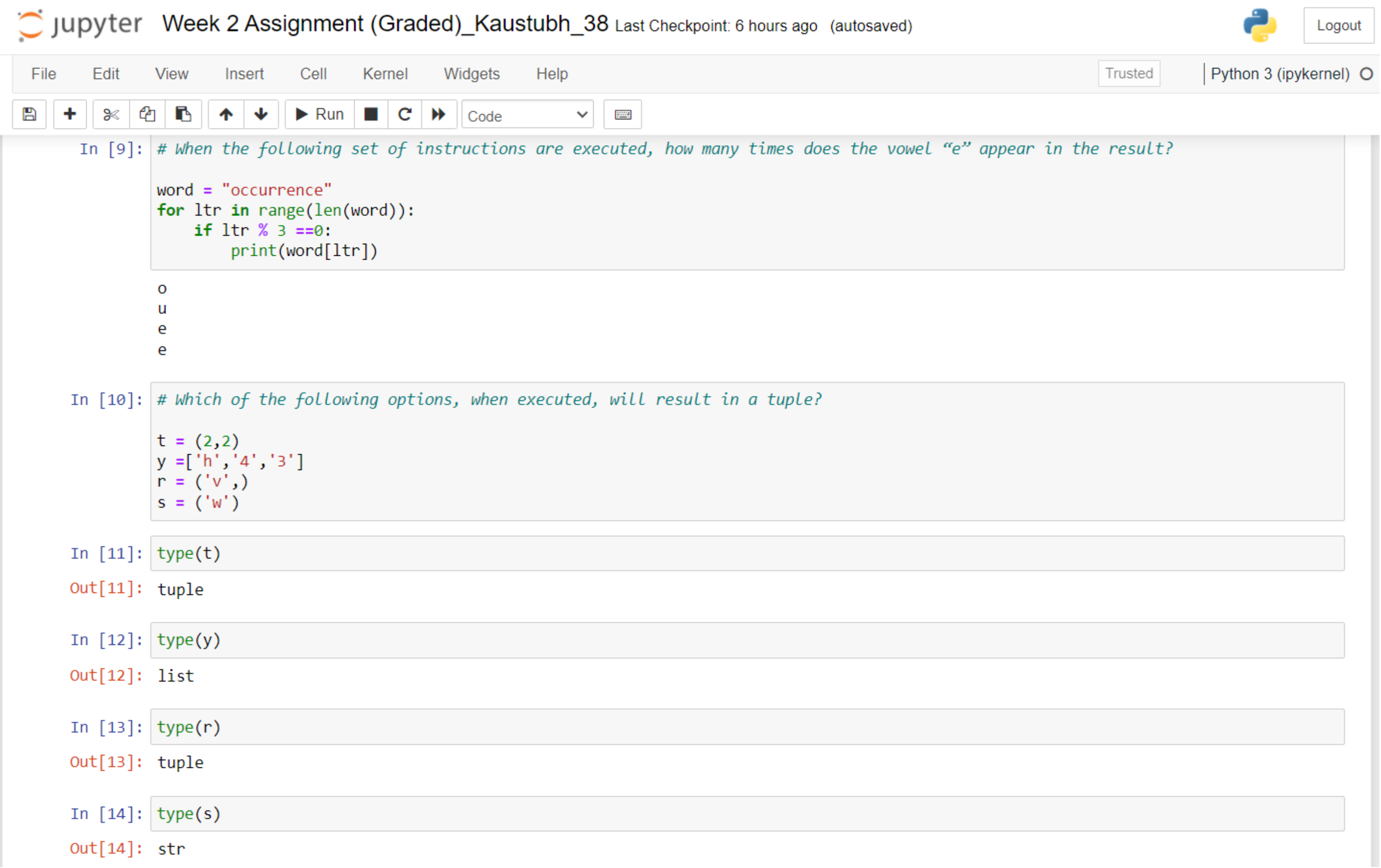
○ 1
○ 1 and 3
○ 1,2,3
○ 3 and 4
● All are correct



```
In [1]: # Consider a variable job = "chemist". Which of the following expressions will
        # retrieve the last character from the variable value?

        job = "chemist"

In [2]: job[7]

        ---------------------------------------------------------------------------
        IndexError                                Traceback (most recent call last)
        ~\AppData\Local\Temp/ipykernel_5592/38642040.py in <module>
        ----> 1 job[7]

        IndexError: string index out of range

In [3]: job[len(job) - 1]

Out[3]: 't'

In [4]: job[5:6]

Out[4]: 's'

In [5]: job[- 1]

Out[5]: 't'
```



```
In [6]: # Which of the following expressions should be used to assign the variable get_num
        # to get the final print statement output as value 75 from the below tuple?

        nst_tup = ("System",(60, 75, 45),(15, 3, 12))
        get_num = nst_tup[1][1]
        print(get_num)

        75

In [7]: # What would be the output for the following set of statements?

        new_list = [13, 23, 18, 64, 51]
        new_list[4] = True
        print(new_list)

        [13, 23, 18, 64, True]

In [8]: # What result does the final statement print?

        scores = (12, 25, 32, 39, 44)
        f_score,*bw_s, l_score = scores
        print("Output is :",f_score,"and",bw_s,"and",l_score)

        Output is : 12 and [25, 32, 39] and 44
```


jupyter
Week 2 Assignment (Graded)_Kaustubh_38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [9]:
```
# When the following set of instructions are executed, how many times does the vowel “e” appear in the result?

word = "occurrence"
for ltr in range(len(word)):
    if ltr % 3 ==0:
        print(word[ltr])
```

```
o
u
e
e
```

In [10]:
```
# Which of the following options, when executed, will result in a tuple?

t = (2,2)
y =['h','4','3']
r = ('v',)
s = ('w')
```

In [11]:
```
type(t)
```

Out[11]:
```
tuple
```

In [12]:
```
type(y)
```

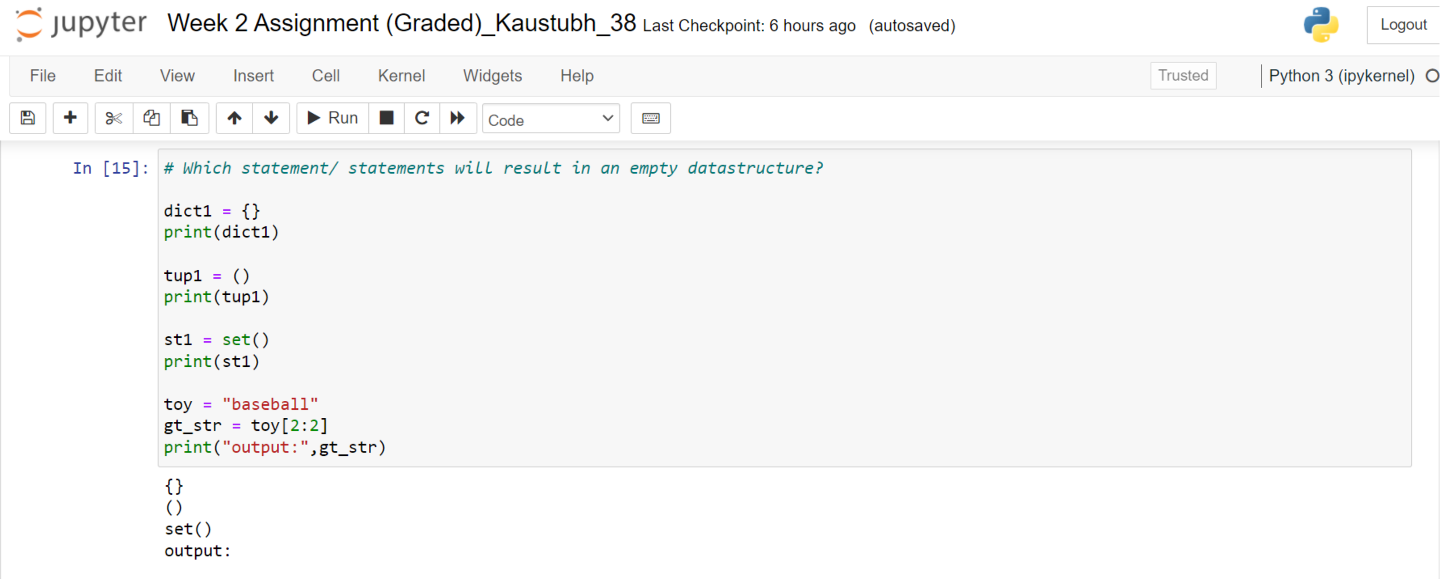
Out[12]:
```
list
```

In [13]:
```
type(r)
```

Out[13]:
```
tuple
```

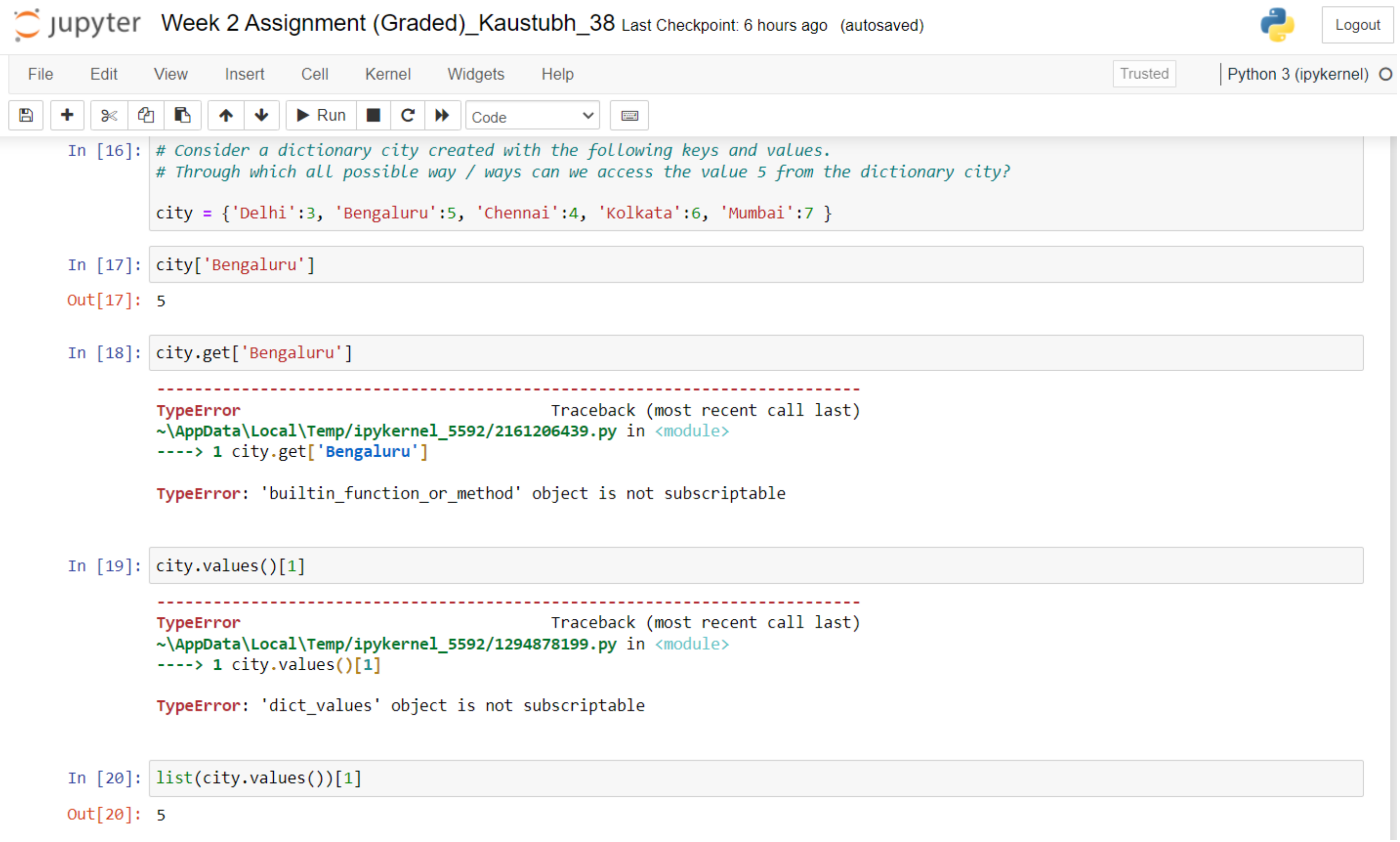
In [14]:
```
type(s)
```

Out[14]:
```
str
```

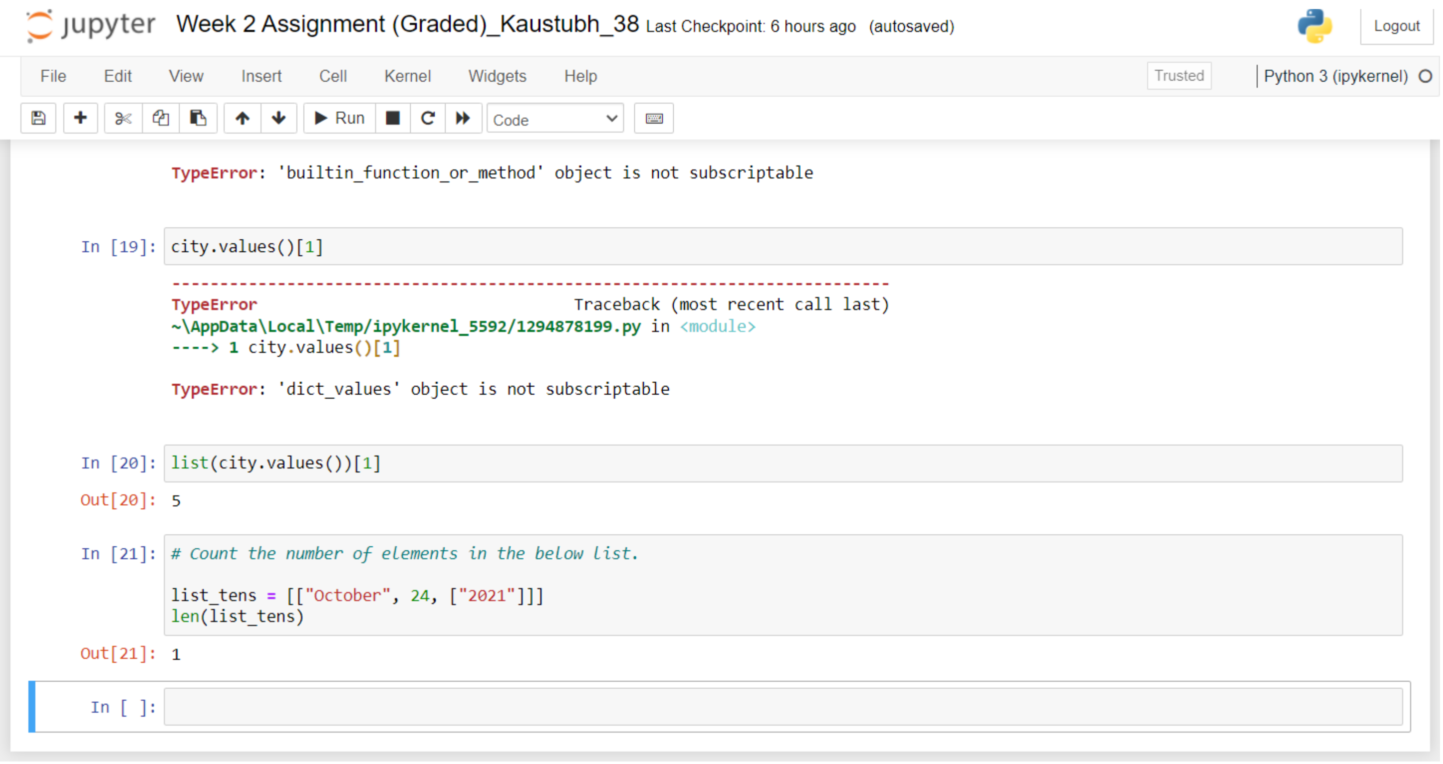

jupyter
Week 2 Assignment (Graded)_Kaustubh_38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [15]:
```
# Which statement/ statements will result in an empty datastructure?

dict1 = {}
print(dict1)

tup1 = ()
print(tup1)

st1 = set()
print(st1)

toy = "baseball"
gt_str = toy[2:2]
print("output:",gt_str)
```

```
{}
()
set()
output:
```



```
In [16]: # Consider a dictionary city created with the following keys and values.
         # Through which all possible way / ways can we access the value 5 from the dictionary city?

         city = {'Delhi':3, 'Bengaluru':5, 'Chennai':4, 'Kolkata':6, 'Mumbai':7 }

In [17]: city['Bengaluru']

Out[17]: 5

In [18]: city.get['Bengaluru']

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         ~\AppData\Local\Temp/ipykernel_5592/2161206439.py in <module>
         ----> 1 city.get['Bengaluru']

         TypeError: 'builtin_function_or_method' object is not subscriptable

In [19]: city.values()[1]

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         ~\AppData\Local\Temp/ipykernel_5592/1294878199.py in <module>
         ----> 1 city.values()[1]

         TypeError: 'dict_values' object is not subscriptable

In [20]: list(city.values())[1]

Out[20]: 5
```



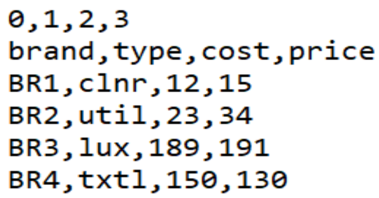
```
         TypeError: 'builtin_function_or_method' object is not subscriptable

In [19]: city.values()[1]

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         ~\AppData\Local\Temp/ipykernel_5592/1294878199.py in <module>
         ----> 1 city.values()[1]

         TypeError: 'dict_values' object is not subscriptable

In [20]: list(city.values())[1]

Out[20]: 5
```

```
In [21]: # Count the number of elements in the below list.

         list_tens = [["October", 24, ["2021"]]]
         len(list_tens)

Out[21]: 1

In [ ]:
```

NPTEL

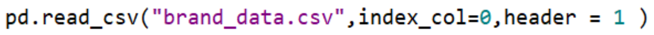
WEEK 3 ASSIGNMENT QUESTIONS

1) Data from the file ***"brand_data.csv "***has to be loaded into a pandas dataframe. A snippet of the data is shown below:

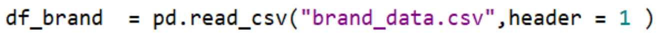


```
0,1,2,3
brand,type,cost,price
BR1,clnr,12,15
BR2,util,23,34
BR3,lux,189,191
BR4,txtl,150,130
```

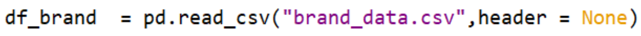
What is the right instruction to read the file into a dataframe df_brand with 4 separate columns?

a) 

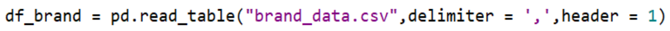
```
pd.read_csv("brand_data.csv",index_col=0,header = 1 )
```

b) 

```
df_brand  = pd.read_csv("brand_data.csv",header = 1 )
```

c) 

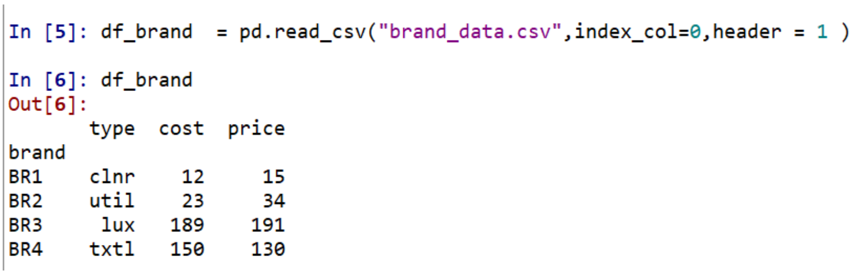
```
df_brand  = pd.read_csv("brand_data.csv",header = None)
```

d) 

```
df_brand = pd.read_table("brand_data.csv",delimiter = ',',header = 1)
```

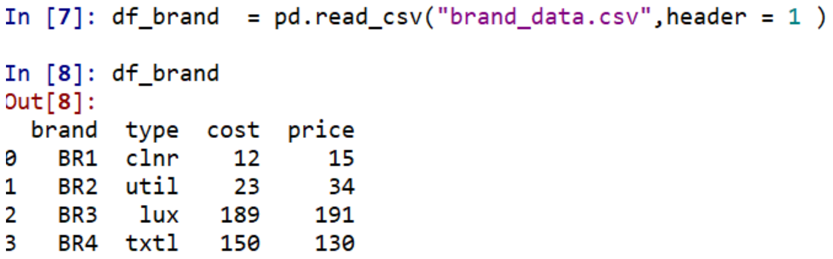
Answers: b) and d)

Option a) chooses the wrong column as index. When set with index_col = 0, the dataframe ends with only 3 columns and brand becomes the index.



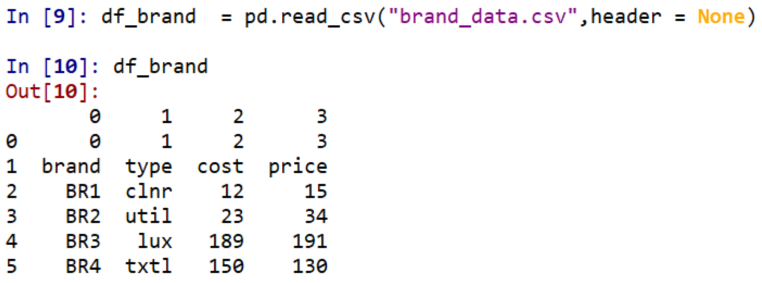
```
In [5]: df_brand  = pd.read_csv("brand_data.csv",index_col=0,header = 1 )

In [6]: df_brand
Out[6]:
       type  cost  price
brand
BR1    clnr    12     15
BR2    util    23     34
BR3     lux   189    191
BR4    txtl   150    130
```

Option b) returns a dataframe of 4 rows and 4 columns. This is correct.



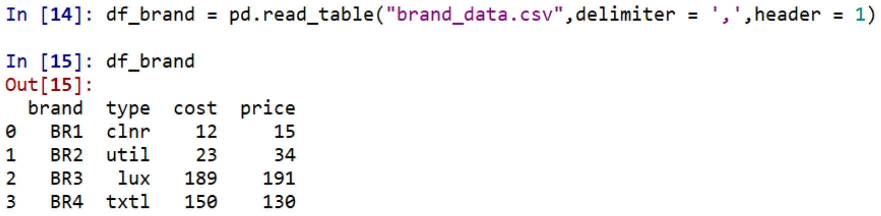
```
In [7]: df_brand  = pd.read_csv("brand_data.csv",header = 1 )

In [8]: df_brand
Out[8]:
  brand  type  cost  price
0   BR1  clnr    12     15
1   BR2  util    23     34
2   BR3   lux   189    191
3   BR4  txtl   150    130
```

Option c) reads the dataframe with the wrong header. The data is read into a dataframe in an illogical manner.



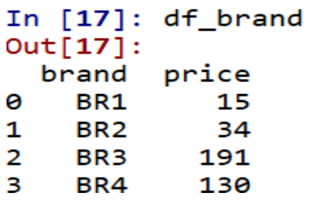
```
In [9]: df_brand  = pd.read_csv("brand_data.csv",header = None)

In [10]: df_brand
Out[10]:
       0     1     2      3
0      0     1     2      3
1  brand  type  cost  price
2    BR1  clnr    12     15
3    BR2  util    23     34
4    BR3   lux   189    191
5    BR4  txtl   150    130
```

Option d) used read_table which can read csv files using the delimiter = ‘,’ setting. Note that the header is also correctly marked. This is correct.



```
In [14]: df_brand = pd.read_table("brand_data.csv",delimiter = ',',header = 1)

In [15]: df_brand
Out[15]:
  brand  type  cost  price
0   BR1  clnr    12     15
1   BR2  util    23     34
2   BR3   lux   189    191
3   BR4  txtl   150    130
```

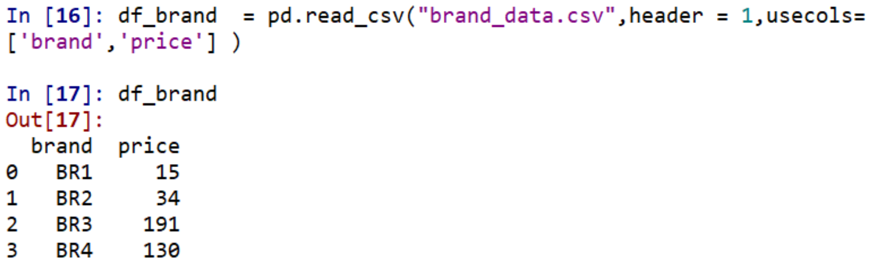
2) For the same file above ***“ brand_data.csv “***, which parameter in pd.read_csv will help to load dataframe df_brand with the selected columns as shown below?



```
In [17]: df_brand
Out[17]:
  brand  price
0   BR1     15
1   BR2     34
2   BR3    191
3   BR4    130
```

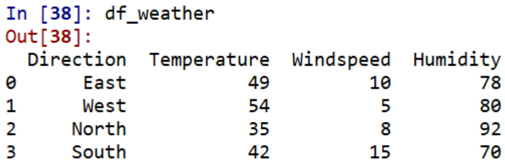
a. index_col =[‘brand’,’Price’]
b. skiprows =[‘brand’,’Price’]
c. usecols =[‘brand’,’Price’]
d. None of the above

Answer: c) usecols. Returns a subset of the columns from the original file.



```
In [16]: df_brand  = pd.read_csv("brand_data.csv",header = 1,usecols=
['brand','price'] )

In [17]: df_brand
Out[17]:
  brand  price
0   BR1     15
1   BR2     34
2   BR3    191
3   BR4    130
```

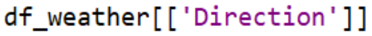
3) Data from the file *“ **weather.xlsx** “* has to be loaded into a pandas dataframe *df_weather* which when printed is as shown below:



```
In [38]: df_weather
Out[38]:
  Direction  Temperature  Windspeed  Humidity
0      East           49         10        78
1      West           54          5        80
2     North           35          8        92
3     South           42         15        70
```

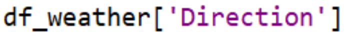
Of the following set of statements which of them can be used to move the column “Direction” into a separate dataframe

a.



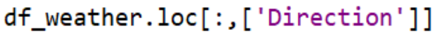
```
df_weather[['Direction']]
```

b.



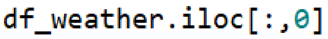
```
df_weather['Direction']
```

c.



```
df_weather.loc[:,['Direction']]
```

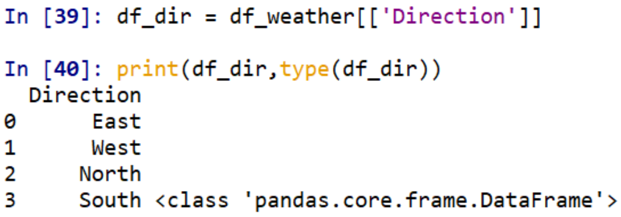
d.



```
df_weather.iloc[:,0]
```

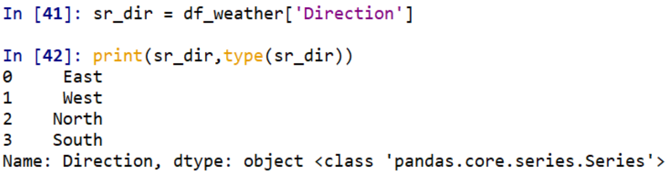
Answer: a and c.

Option a. ->



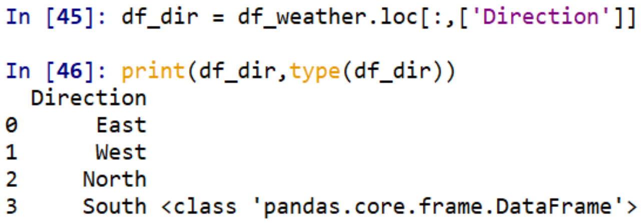
```
In [39]: df_dir = df_weather[['Direction']]

In [40]: print(df_dir,type(df_dir))
  Direction
0      East
1      West
2     North
3     South <class 'pandas.core.frame.DataFrame'>
```

Option b ->



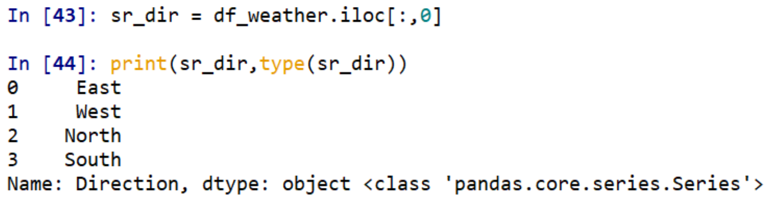
```
In [41]: sr_dir = df_weather['Direction']

In [42]: print(sr_dir,type(sr_dir))
0     East
1     West
2    North
3    South
Name: Direction, dtype: object <class 'pandas.core.series.Series'>
```

Option c ->



```
In [45]: df_dir = df_weather.loc[:,['Direction']]

In [46]: print(df_dir,type(df_dir))
  Direction
0      East
1      West
2     North
3     South <class 'pandas.core.frame.DataFrame'>
```

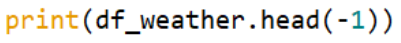
Option d ->



```
In [43]: sr_dir = df_weather.iloc[:,0]

In [44]: print(sr_dir,type(sr_dir))
0     East
1     West
2    North
3    South
Name: Direction, dtype: object <class 'pandas.core.series.Series'>
```

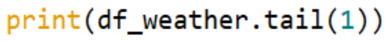
4) Referring to the same dataframe df_weather in Question (3), which statement/statements will help to print the last row from the dataframe?

a.



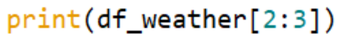
```
print(df_weather.head(-1))
```

b.



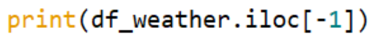
```
print(df_weather.tail(1))
```

c.



```
print(df_weather[2:3])
```

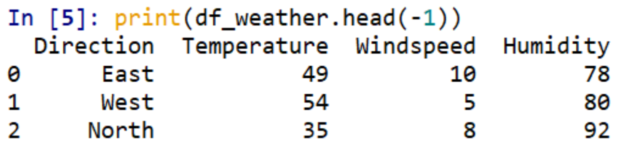
d.



```
print(df_weather.iloc[-1])
```

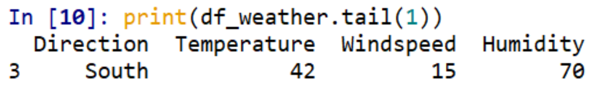
Answer: b and d

Option a. Retrieves all rows except the last row.



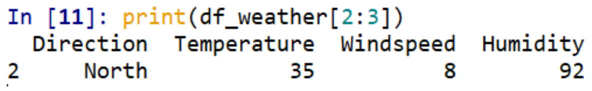
```
In [5]: print(df_weather.head(-1))
  Direction  Temperature  Windspeed  Humidity
0      East           49         10        78
1      West           54          5        80
2     North           35          8        92
```

Option b. Correct option.



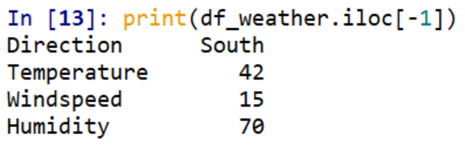
```
In [10]: print(df_weather.tail(1))
  Direction  Temperature  Windspeed  Humidity
3     South           42         15        70
```

Option c. Retrieves the row with index 2 [second last row].



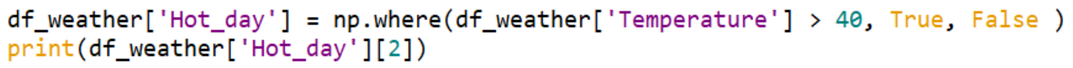
```
In [11]: print(df_weather[2:3])
  Direction  Temperature  Windspeed  Humidity
2     North           35          8        92
```

Option d. Correct option.



```
In [13]: print(df_weather.iloc[-1])
Direction      South
Temperature       42
Windspeed         15
Humidity          70
```

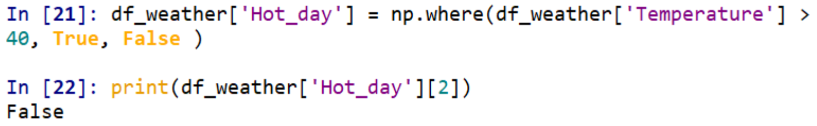
5) In reference to the same dataframe df_weather, we add an additional column 'Hot_day' to determine whether the day is hot or not based on the values in the Temperature column. What will the print statement derive?



```
df_weather['Hot_day'] = np.where(df_weather['Temperature'] > 40, True, False )
print(df_weather['Hot_day'][2])
```

a. True
b. SyntaxError
c. False
d. None of the above

Answer: c). The third row has a temperature of 35, so it will return False.



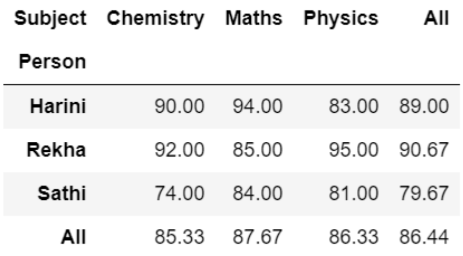
```
In [21]: df_weather['Hot_day'] = np.where(df_weather['Temperature'] >
40, True, False )

In [22]: print(df_weather['Hot_day'][2])
False
```

6) What statement would give the number of columns in a dataframe df?

a. len(df.columns)
b. len(df)
c. df.size
d. All of the above.

Answer: a) len(df) returns number of rows. df.size returns the number of elements.

7) A file **"Students.csv"** contains the attendance and total scores of three separate students. This data is loaded into a dataframe **df_study** and a pandas crosstab is applied on the same dataframe which results in the following output



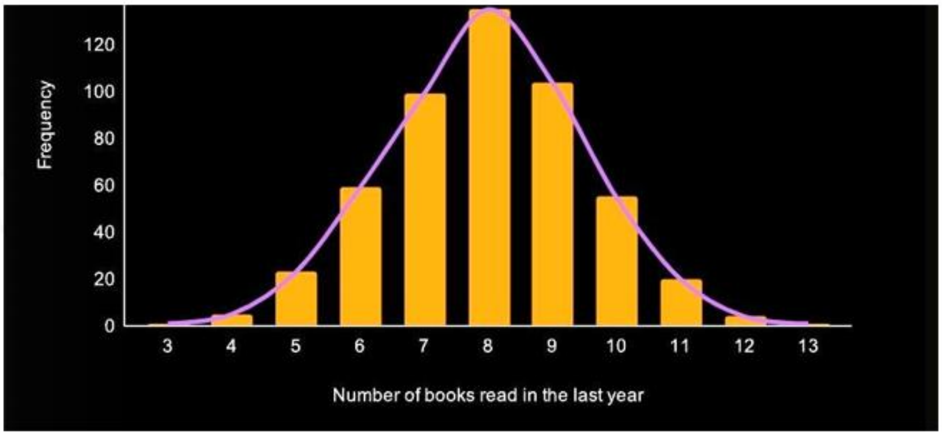
Subject	Chemistry	Maths	Physics	All
Person				
Harini	90.00	94.00	83.00	89.00
Rekha	92.00	85.00	95.00	90.67
Sathi	74.00	84.00	81.00	79.67
All	85.33	87.67	86.33	86.44

Which student scored the maximum average score of all three subjects? Which subject has the best average score for all three students?

a. Harini,Chemistry
b. Rekha,Physics
c. Harini,Physics
d. Rekha,Maths

Answer: d) Rekha, Maths.

8) The following histogram shows the number of books read in a year:

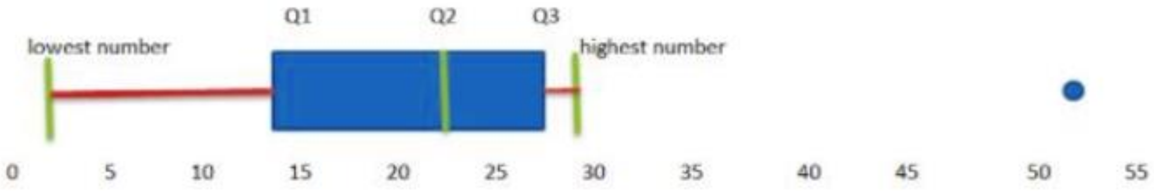


Find the mean and median in the above histogram.

a) 7,8
b) 8,9
c) 8.5,7
d) 8,8
e) None of the above

Answer: d) 8 is the central tendency for the above histogram. It is the mean, median and mode.

9) For the following box plot, which among the given options are the median and the outlier?

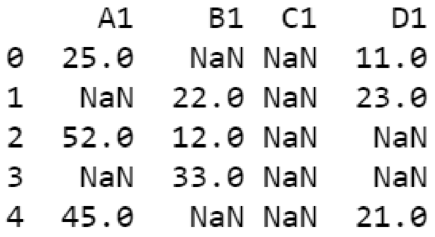


a. 15,52
b. 22, 52
c. 13.5, 29
d. 25, 50

Answer: b) Median is between 20 and 25, so 22 is the median. Outlier is between 50 and 55, hence 52 is the outlier.

Q1 -13.5   Q3 – 27.5

10) A dataframe df_logs has the following data.



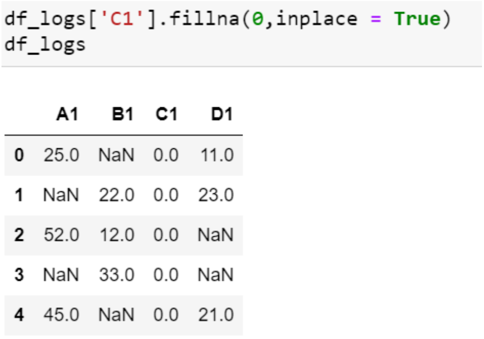
```
     A1    B1  C1    D1
0  25.0   NaN NaN  11.0
1   NaN  22.0 NaN  23.0
2  52.0  12.0 NaN   NaN
3   NaN  33.0 NaN   NaN
4  45.0   NaN NaN  21.0
```

All the NaN / Null values in the column C1 can be replaced by zero value by executing which of the following statements?

a. df_logs['C1'].fillna(0,inplace = True)
b. df_logs.fillna(0,inplace = True)
c. df_logs.fillna(0,inplace = False)
d. df_logs['C1'].fillna(df_logs['B1'],inplace = True)

a. Answer: a) df_logs['C1'].fillna(0,inplace = True)

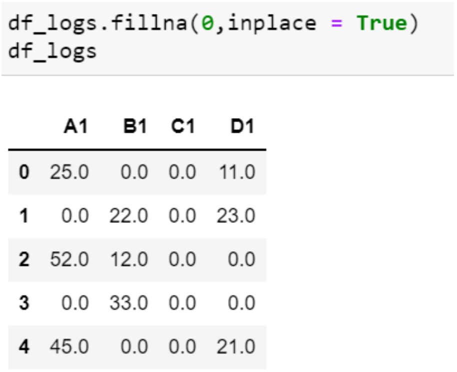
Option a) Only Column C1 values get replaced by zero value.



```
df_logs['C1'].fillna(0,inplace = True)
df_logs
```

	A1	B1	C1	D1
0	25.0	NaN	0.0	11.0
1	NaN	22.0	0.0	23.0
2	52.0	12.0	0.0	NaN
3	NaN	33.0	0.0	NaN
4	45.0	NaN	0.0	21.0

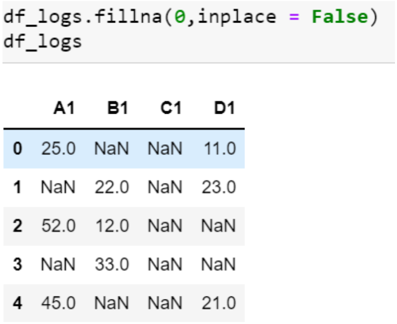
Option b). All the null values in the dataframe get replaced by zero value. Incorrect.



```
df_logs.fillna(0,inplace = True)
df_logs
```

	A1	B1	C1	D1
0	25.0	0.0	0.0	11.0
1	0.0	22.0	0.0	23.0
2	52.0	12.0	0.0	0.0
3	0.0	33.0	0.0	0.0
4	45.0	0.0	0.0	21.0

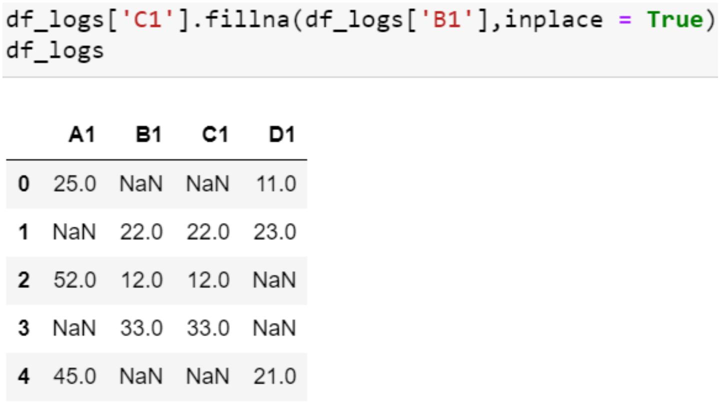
Option c). No changes are reflected in the dataframe. Incorrect.



```
df_logs.fillna(0,inplace = False)
df_logs
```

	A1	B1	C1	D1
0	25.0	NaN	NaN	11.0
1	NaN	22.0	NaN	23.0
2	52.0	12.0	NaN	NaN
3	NaN	33.0	NaN	NaN
4	45.0	NaN	NaN	21.0

Option d). Column C1 null values get replaced by Column B1 values. Incorrect.



```
df_logs['C1'].fillna(df_logs['B1'],inplace = True)
df_logs
```

	A1	B1	C1	D1
0	25.0	NaN	NaN	11.0
1	NaN	22.0	22.0	23.0
2	52.0	12.0	12.0	NaN
3	NaN	33.0	33.0	NaN
4	45.0	NaN	NaN	21.0

NPTEL

WEEK 4 ASSIGNMENT QUESTIONS

Given Data: Credit Worthiness data containing 1000 observations of income details of individuals comprising 21 attributes along the columns (Cbal, Cdur, Chist, Cpur, Camt, Sbal, Edur, InRate, MSG, Oparties, Rdur, Prop, age, inPlans, Htype, NumCred, JobType, Ndepend, telephone, foreign, creditScore)
Problem statement: By observing the features of the dataset, the problem statement can be defined as a binary classification problem of classifying any individual into an appropriate category of creditScore such as Good or Bad.

1) How many unique values are present in the Sbal feature; also, what is the most frequent value within Sbal?

a) 5, Rs. >= 10,000

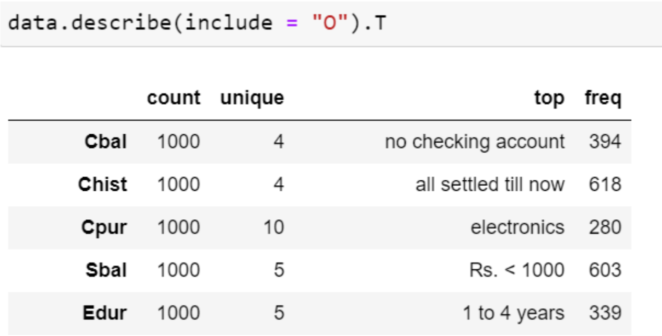
b) 4, Rs. < 1000

c) 5, Rs. < 1000

d) 4, '1000 <= Rs. < 5,000'

Answers: c)

All features of object type can be analyzed by describe (). MARKUP ON THE PICTURE.



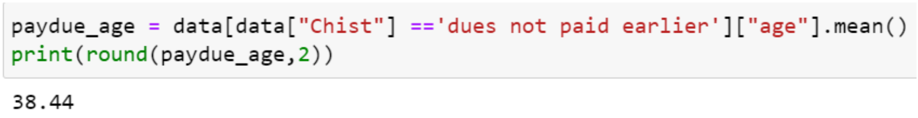
```
data.describe(include = "O").T
```

	count	unique	top	freq
Cbal	1000	4	no checking account	394
Chist	1000	4	all settled till now	618
Cpur	1000	10	electronics	280
Sbal	1000	5	Rs. < 1000	603
Edur	1000	5	1 to 4 years	339

2) Find the average age of those customers who have a credit history [Chist] wherein the dues are not paid earlier.

a. 35.54
b. 38.44
c. 33.00
d. None of the above

Answer: b) 38.44



```
paydue_age = data[data["Chist"] =='dues not paid earlier']["age"].mean()
print(round(paydue_age,2))
```

```
38.44
```

3) A Logistic Regression model is built in which none of the features used are standardized. The train to test proportion is 75:25 and the random state is set to 1. The accuracy of the model is ________.

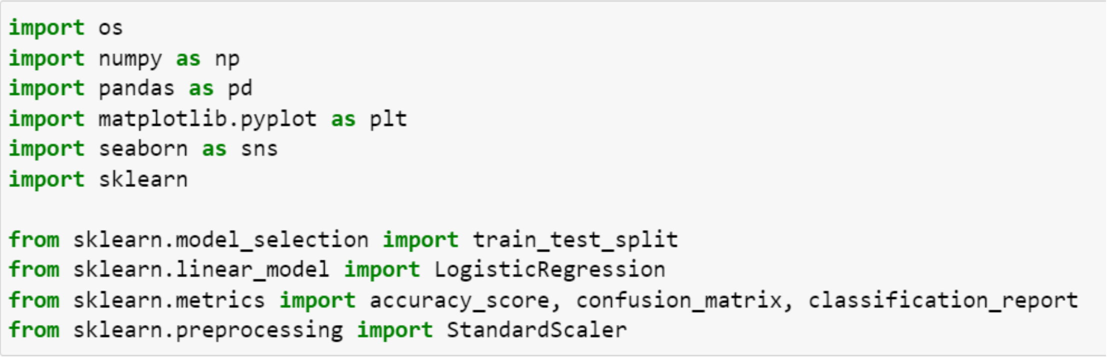
a. Less than 50%

b. Between 50% and 60%

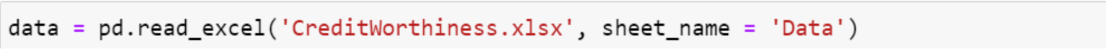
c. Greater than 70%

d. None of the above

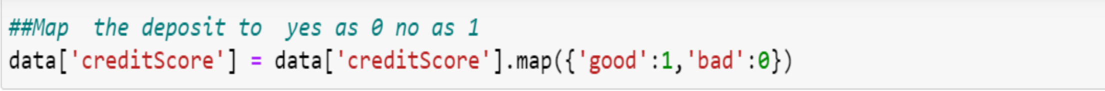
Answer: c)



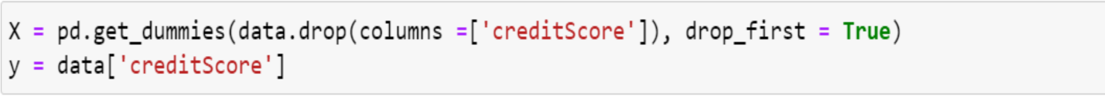
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
```



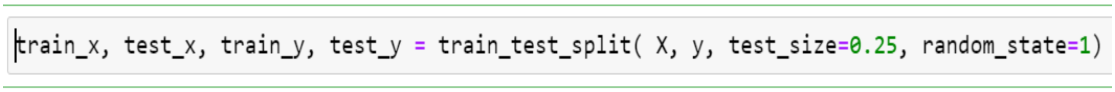
```
data = pd.read_excel('CreditWorthiness.xlsx', sheet_name = 'Data')
```



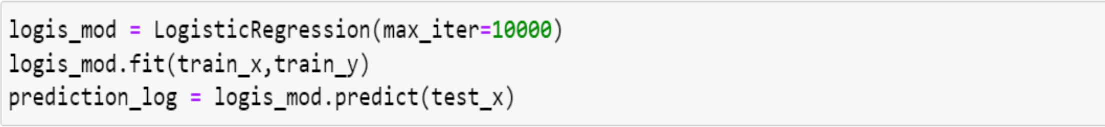
```
##Map  the deposit to  yes as 0 no as 1
data['creditScore'] = data['creditScore'].map({'good':1,'bad':0})
```



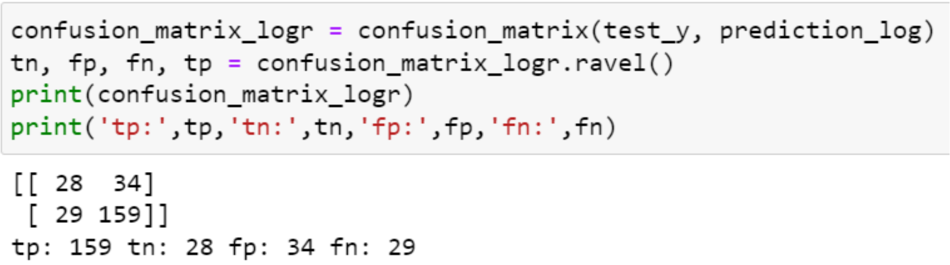
```
X = pd.get_dummies(data.drop(columns =['creditScore']), drop_first = True)
y = data['creditScore']
```



```
train_x, test_x, train_y, test_y = train_test_split( X, y, test_size=0.25, random_state=1)
```

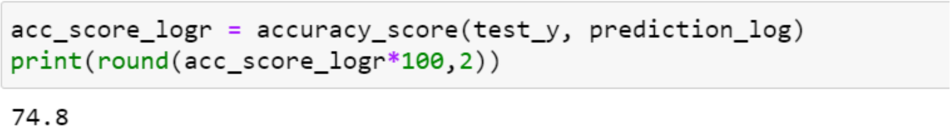


```
logis_mod = LogisticRegression(max_iter=10000)
logis_mod.fit(train_x,train_y)
prediction_log = logis_mod.predict(test_x)
```



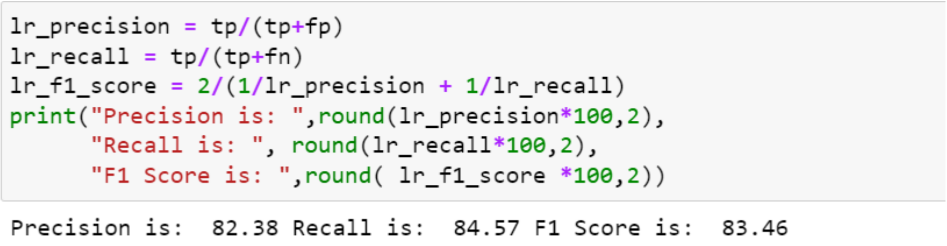
```
confusion_matrix_logr = confusion_matrix(test_y, prediction_log)
tn, fp, fn, tp = confusion_matrix_logr.ravel()
print(confusion_matrix_logr)
print('tp:',tp,'tn:',tn,'fp:',fp,'fn:',fn)
```

```
[[ 28  34]
 [ 29 159]]
tp: 159 tn: 28 fp: 34 fn: 29
```



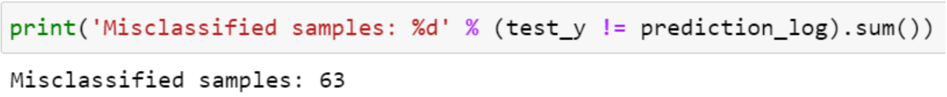
```
acc_score_logr = accuracy_score(test_y, prediction_log)
print(round(acc_score_logr*100,2))
```

```
74.8
```



```
lr_precision = tp/(tp+fp)
lr_recall = tp/(tp+fn)
lr_f1_score = 2/(1/lr_precision + 1/lr_recall)
print("Precision is: ",round(lr_precision*100,2),
      "Recall is: ", round(lr_recall*100,2),
      "F1 Score is: ",round( lr_f1_score *100,2))
```

```
Precision is:  82.38 Recall is:  84.57 F1 Score is:  83.46
```



```
print('Misclassified samples: %d' % (test_y != prediction_log).sum())
```

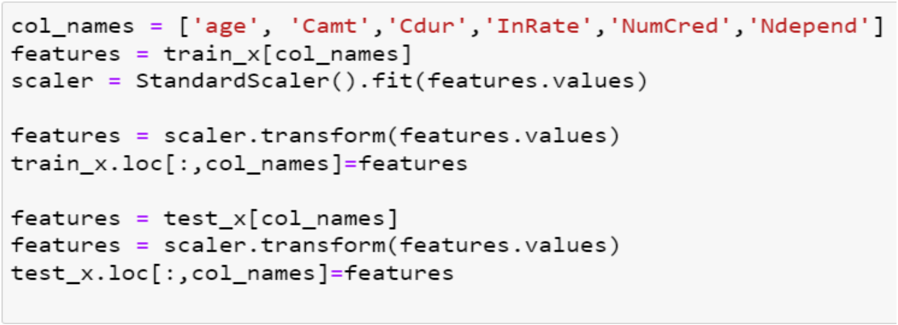
```
Misclassified samples: 63
```

4) Import StandardScaler() from the sklearn.preprocessing package to standardize the features. Use the same train-test proportion and the random state should be set to 1. After standardizing the logistic regression model, by what percentage has the misclassified samples changed?

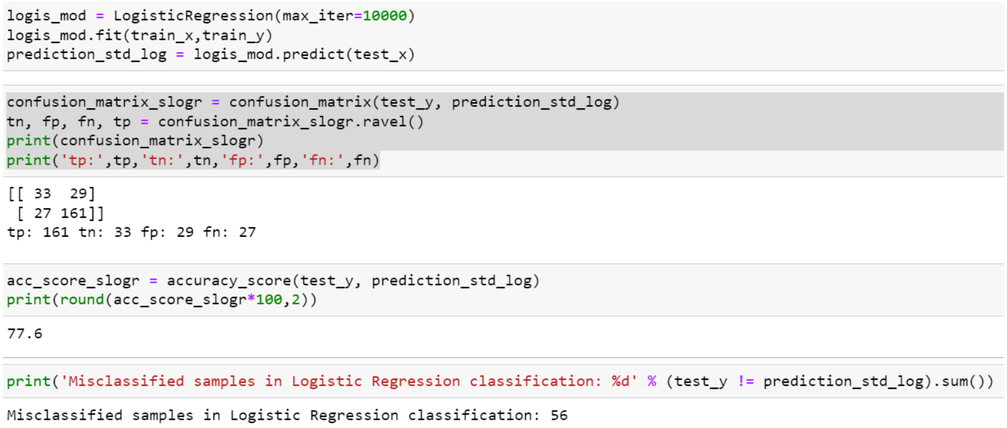
a. 11.11%
b. 3.7%
c. 20%
d. 39.2%

Answer: a

After Standardizing:



```
col_names = ['age', 'Camt','Cdur','InRate','NumCred','Ndepend']
features = train_x[col_names]
scaler = StandardScaler().fit(features.values)

features = scaler.transform(features.values)
train_x.loc[:,col_names]=features

features = test_x[col_names]
features = scaler.transform(features.values)
test_x.loc[:,col_names]=features
```



```
logis_mod = LogisticRegression(max_iter=10000)
logis_mod.fit(train_x,train_y)
prediction_std_log = logis_mod.predict(test_x)
```

```
confusion_matrix_slogr = confusion_matrix(test_y, prediction_std_log)
tn, fp, fn, tp = confusion_matrix_slogr.ravel()
print(confusion_matrix_slogr)
print('tp:',tp,'tn:',tn,'fp:',fp,'fn:',fn)
```

```
[[ 33  29]
 [ 27 161]]
tp: 161 tn: 33 fp: 29 fn: 27
```

```
acc_score_slogr = accuracy_score(test_y, prediction_std_log)
print(round(acc_score_slogr*100,2))
```

```
77.6
```

```
print('Misclassified samples in Logistic Regression classification: %d' % (test_y != prediction_std_log).sum())
```

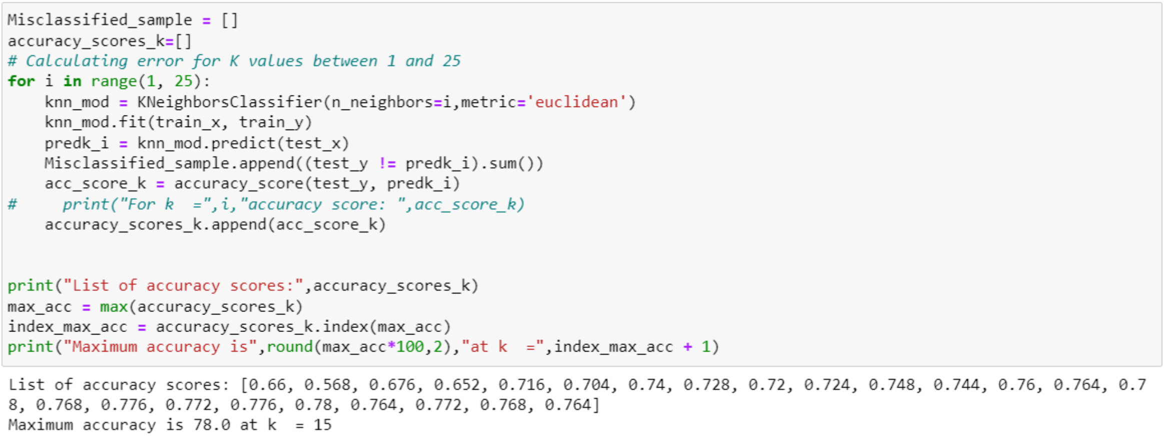
```
Misclassified samples in Logistic Regression classification: 56
```

Percentage change in misclassified samples : (56-63 /63)*100 = 11.11%

5) When KNN classification is applied on the same standardized data at the optimal value for k nearest neighbours, the accuracy achieved is _______.

a. 64%
b. 78%
c. 76.4%
d. None of the above

Answer: b)



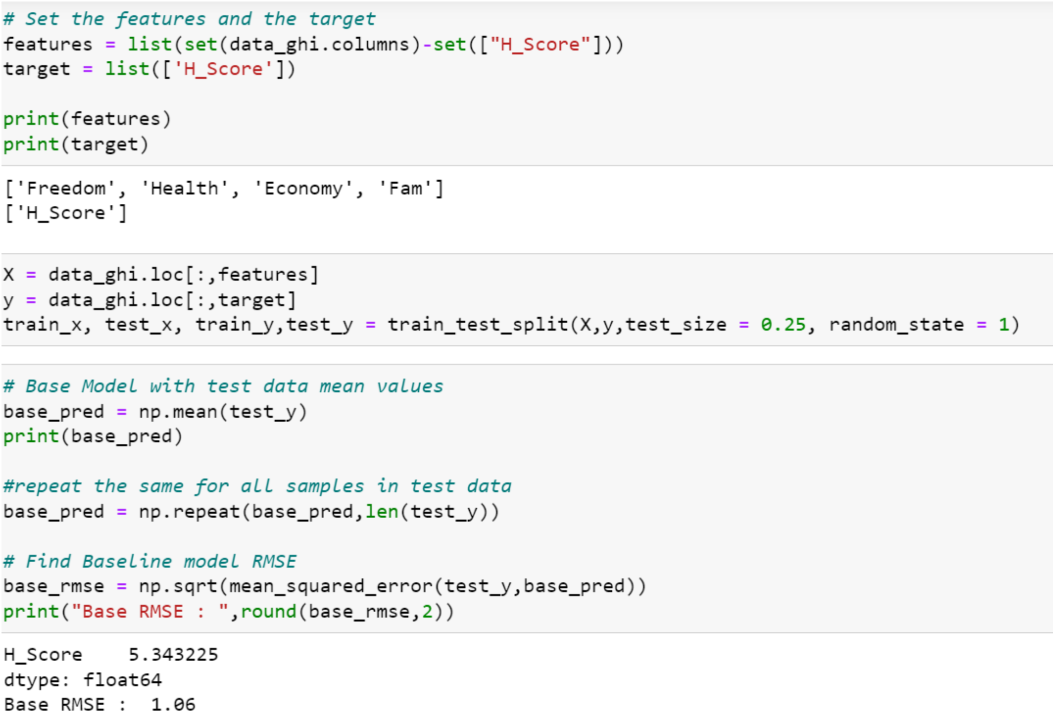
```
Misclassified_sample = []
accuracy_scores_k=[]
# Calculating error for K values between 1 and 25
for i in range(1, 25):
    knn_mod = KNeighborsClassifier(n_neighbors=i,metric='euclidean')
    knn_mod.fit(train_x, train_y)
    predk_i = knn_mod.predict(test_x)
    Misclassified_sample.append((test_y != predk_i).sum())
    acc_score_k = accuracy_score(test_y, predk_i)
#     print("For k  =",i,"accuracy score: ",acc_score_k)
    accuracy_scores_k.append(acc_score_k)


print("List of accuracy scores:",accuracy_scores_k)
max_acc = max(accuracy_scores_k)
index_max_acc = accuracy_scores_k.index(max_acc)
print("Maximum accuracy is",round(max_acc*100,2),"at k  =",index_max_acc + 1)
```

```
List of accuracy scores: [0.66, 0.568, 0.676, 0.652, 0.716, 0.704, 0.74, 0.728, 0.72, 0.724, 0.748, 0.744, 0.76, 0.764, 0.7
8, 0.768, 0.776, 0.772, 0.776, 0.78, 0.764, 0.772, 0.768, 0.764]
Maximum accuracy is 78.0 at k  = 15
```

6) A multiple linear regression model is built on the Global Happiness Index dataset "GHI_Report.csv". What is the rmse of the baseline model?

a. 1.99
b. 0.85
c. 1.06
d. 0.33

Answer: b) 1.06



```
# Set the features and the target
features = list(set(data_ghi.columns)-set(["H_Score"]))
target = list(['H_Score'])

print(features)
print(target)
```

```
['Freedom', 'Health', 'Economy', 'Fam']
['H_Score']
```

```
X = data_ghi.loc[:,features]
y = data_ghi.loc[:,target]
train_x, test_x, train_y,test_y = train_test_split(X,y,test_size = 0.25, random_state = 1)
```

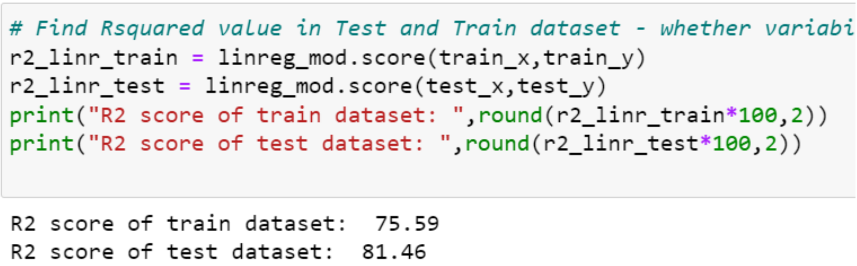
```
# Base Model with test data mean values
base_pred = np.mean(test_y)
print(base_pred)

#repeat the same for all samples in test data
base_pred = np.repeat(base_pred,len(test_y))

# Find Baseline model RMSE
base_rmse = np.sqrt(mean_squared_error(test_y,base_pred))
print("Base RMSE : ",round(base_rmse,2))
```

```
H_Score    5.343225
dtype: float64
Base RMSE :  1.06
```

7) From the multiple linear regression model built on the GHI index, we get an R-squared value of ____ on the test data subset.

a. 55.63
b. 45.81
c. 75.59
d. 81.46

Answer: d)



```
# Find Rsquared value in Test and Train dataset - whether variabi
r2_linr_train = linreg_mod.score(train_x,train_y)
r2_linr_test = linreg_mod.score(test_x,test_y)
print("R2 score of train dataset: ",round(r2_linr_train*100,2))
print("R2 score of test dataset: ",round(r2_linr_test*100,2))
```

```
R2 score of train dataset:  75.59
R2 score of test dataset:  81.46
```

8) Which of the following statement/s about Linear Regression is / are true?

a) Linear Regression assumes that there exists a linear relationship between the independent variable and dependent variable.
b) The errors terms are assumed to be independent and normally distributed.
c) The percentage of variation in the dependent variable as explained by the independent variable/variables is expressed by R-squared value.
d) Residuals are the product of the predicted value and the actual observed value.

Answer: a,b and c.

Residuals are the difference between the predicted value and the actual observed value.

9) Which of the following statements is inaccurate about Logistic Regression?

a) Logistic Regression doesn't require a linear relationship between the dependent and independent variables.
b) The value of the logistic function being a probability will range between 0 and 1.
c) Cost function of Logistic Regression is also called as the Log Loss function.
d) The dependent variable can be of both numerical or categorical type just like the independent variables.

Answer: d) Only categorical dependent variable.

10) In a KNN model, by which means do we handle categorical variables?

a) Standardization
b) Dummy variables
c) Correlation
d) None of the above

Answer: b) Dummy variables can be used to encode the different values contained in a particular categorical independent feature.