Section A

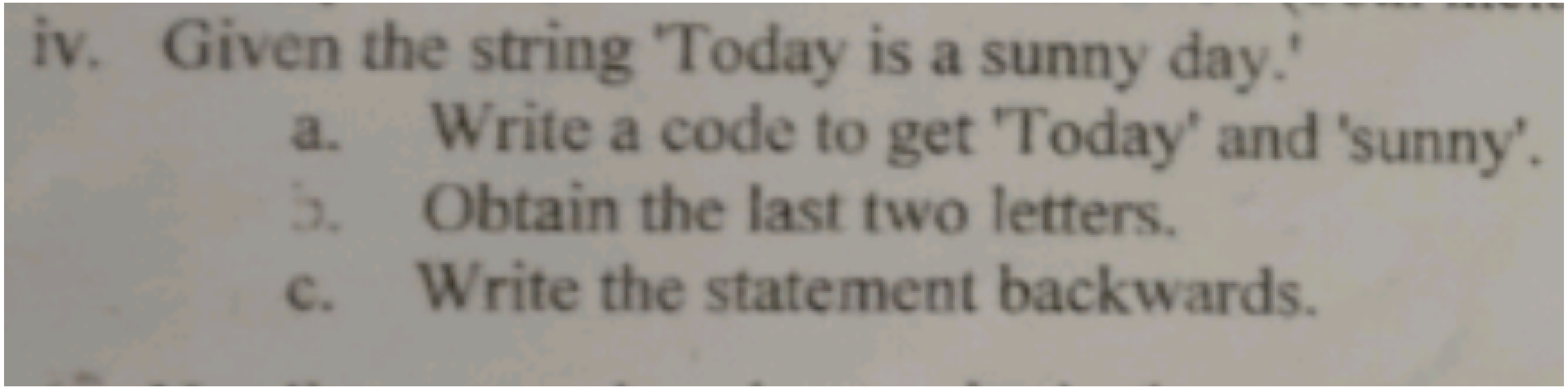
Q1(iii)

In [1]:

```
result=[i for i in range(1500,2701) if i%5==0 and i%7==0]
ans=','.join(map(str,result))
print(ans)
```

```
1505,1540,1575,1610,1645,1680,1715,1750,1785,1820,1855,1890,1925,1960,1995,2030,2065,2100,2135,2170,2205,2240,2275,2310,2345,2380,2415,245
0,2485,2520,2555,2590,2625,2660,2695
```

Q1(iv)



iv. Given the string 'Today is a sunny day.'

a. Write a code to get 'Today' and 'sunny'.

b. Obtain the last two letters.

c. Write the statement backwards.

In [2]:

```
String="Today is a sunny day"
#Ans (a)
String=String.split()
print("Answer is : ",String[0:4:3])
#Ans (b)
print("Answer is : ",String[-1:-3:-1])
#Ans (c)
print("Reversed : ",' '.join(String[::-1]))
```

```
Answer is :  ['Today', 'sunny']
Answer is :  ['day', 'sunny']
Reversed :  day sunny a is Today
```

Q1(V) Use list comprehension to obtain the square root of first 10 natural numbers.

In [3]:

```
import math
result=[math.sqrt(i) for i in range(1,11)]
result=', '.join(map(lambda x: str(format(x,".2f")),result))
print(result)
```

```
1.00, 1.41, 1.73, 2.00, 2.24, 2.45, 2.65, 2.83, 3.00, 3.16
```

Q1(vi) How to filter words that contain at least 2 vowels from the series

sr=pd.Series(["Apple","Orange","Plan","Python","Money"]

In [4]:

```
#Way 1
import pandas as pd
print(list(filter(lambda x:x if sum(list(map(lambda a: 1 if a in "aeiouAEIOU" else 0,x)))>1 else None,pd.Series(["Apple","Orange","Plan","P
```

```
['Apple', 'Orange', 'Money']
```

In [5]:

```
#way 2
def verify(string):
    if (sum(list(map(lambda a: 1 if a in "aeiouAEIOU" else 0,string)))>1):
        return string
print(list(filter(verify,pd.Series(["Apple","Orange","Plan","Python","Money"]))))
```

```
['Apple', 'Orange', 'Money']
```

Q1(vii)

Write a code to find first and last five rows in a dataset

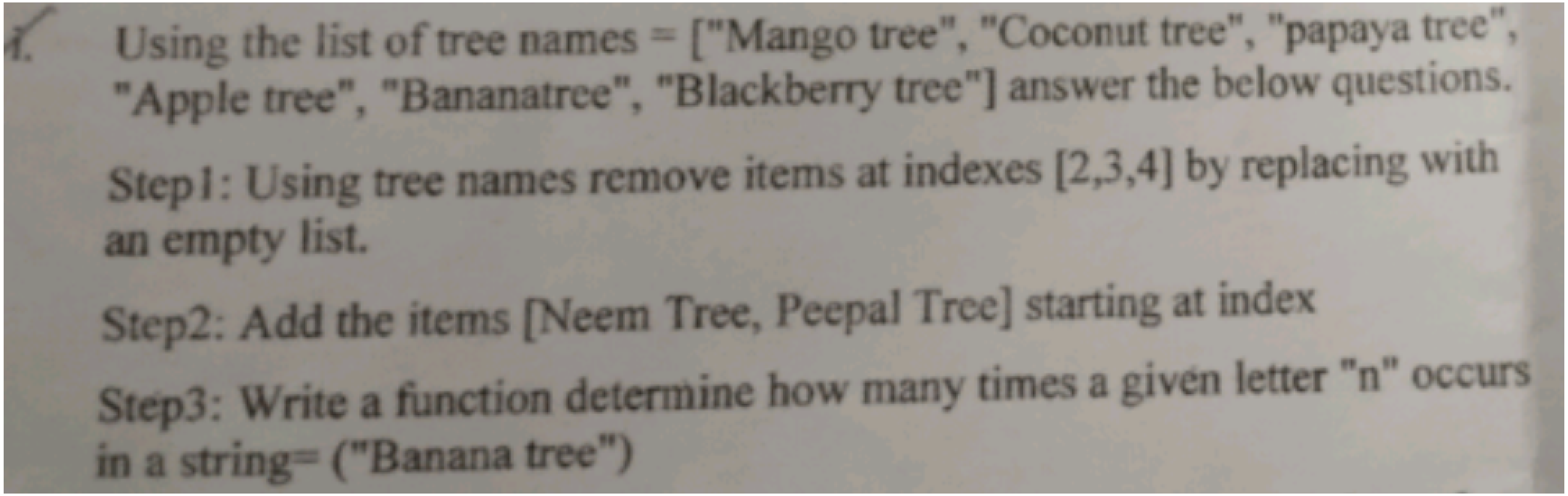
In [44]:

```
#Data set
import pandas as pd
data={ "ids" : [1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010],
        "Name": ["Ardhana","Mehul","Rishabh","Vijay","Alice", "Bob", "Charlie", "David", "Emma", "Frank"],
        "designations" : ["Manager", "Engineer", "Analyst", "Director", "Coordinator", "Developer", "Supervisor", "Specialist", "Consultan
        "salaries" : [50000, 60000, 75000, 90000, 55000, 80000, 65000, 70000, 85000, 95000]
       }
dataset=pd.DataFrame(data,index=[1,2,3,4,5,6,7,8,9,10])
print("______________Data Set________________")
print(dataset)
print("______________First Five Rows________________")
print(dataset.head(5))
print("______________last Five Rows_________________")
print(dataset.tail(5))
```

```
______________Data Set________________
     ids     Name   designations  salaries
1   1001  Ardhana        Manager     50000
2   1002    Mehul       Engineer     60000
3   1003  Rishabh        Analyst     75000
4   1004    Vijay       Director     90000
5   1005    Alice    Coordinator     55000
6   1006      Bob      Developer     80000
7   1007  Charlie     Supervisor     65000
8   1008    David     Specialist     70000
9   1009     Emma     Consultant     85000
10  1010    Frank  Administrator     95000
______________First Five Rows________________
    ids     Name designations  salaries
1  1001  Ardhana      Manager     50000
2  1002    Mehul     Engineer     60000
3  1003  Rishabh      Analyst     75000
4  1004    Vijay     Director     90000
5  1005    Alice  Coordinator     55000
______________last Five Rows_________________
     ids     Name   designations  salaries
6   1006      Bob      Developer     80000
7   1007  Charlie     Supervisor     65000
8   1008    David     Specialist     70000
9   1009     Emma     Consultant     85000
10  1010    Frank  Administrator     95000
```

Section - B

(i)



1. Using the list of tree names = ["Mango tree", "Coconut tree", "papaya tree", "Apple tree", "Bananatree", "Blackberry tree"] answer the below questions.

Step1: Using tree names remove items at indexes [2,3,4] by replacing with an empty list.

Step2: Add the items [Neem Tree, Peepal Tree] starting at index

Step3: Write a function determine how many times a given letter "n" occurs in a string= ("Banana tree")

In [51]:

```
#step 1
trees=["Mango tree","Coconut tree","Papaya tree","Apple tree","Banana tree","Blackberry tree"]
trees.remove(trees[trees.index("Papaya tree")])
trees.remove(trees[trees.index("Apple tree")])
trees.remove(trees[trees.index("Banana tree")])
print(trees)
```

```
['Mango tree', 'Coconut tree', 'Apple tree']
```

In [52]:

```
#step 2
new_trees=["Neem Tree","Peepal Tree"]
trees[:0]=new_trees
print(trees)
```

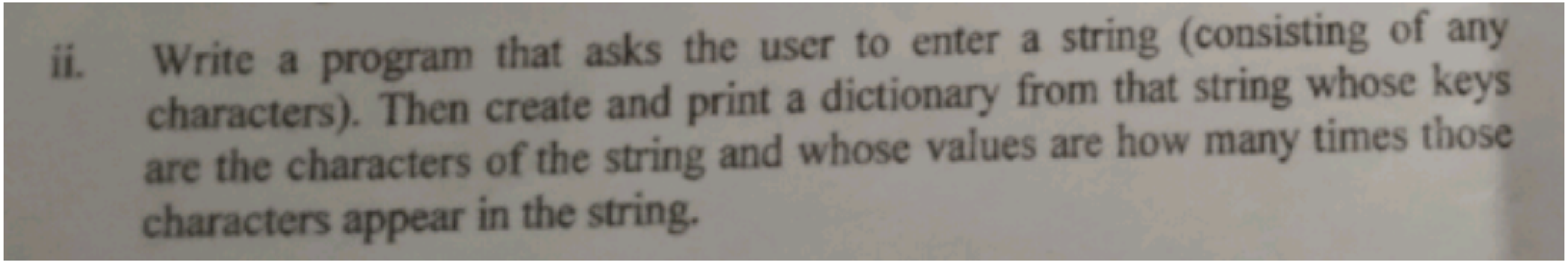
```
['Neem Tree', 'Peepal Tree', 'Mango tree', 'Coconut tree', 'Apple tree']
```

In [6]:

```
#Step 3
ans = sum(map(lambda x: True if x == 'n' else False, 'Banana tree'))
print(ans)
```

```
2
```

(ii)

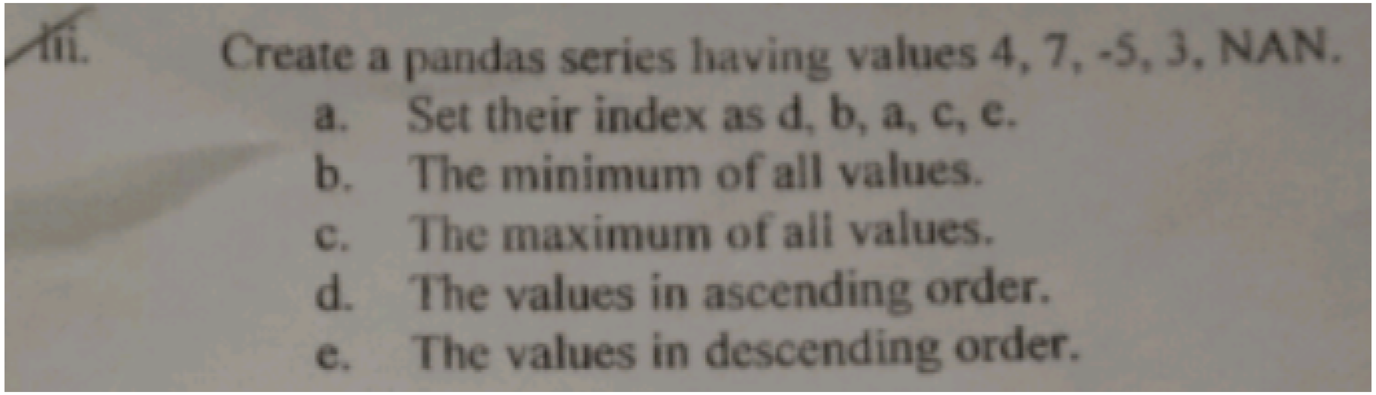


ii. Write a program that asks the user to enter a string (consisting of any characters). Then create and print a dictionary from that string whose keys are the characters of the string and whose values are how many times those characters appear in the string.

In [2]:

```
#Answer
string=list(input("Enter a string : "))
keys=set(string)
values=[string.count(i) for i in keys]
occurrence=dict(zip(keys,values))
for x,y in occurrence.items():
    print(x,y)
```

```
Enter a string : AAACCCCBBBB######**
B 4
C 4
* 2
A 3
# 6
```



~~iii.~~ Create a pandas series having values 4, 7, -5, 3, NAN.

a. Set their index as d, b, a, c, e.
b. The minimum of all values.
c. The maximum of all values.
d. The values in ascending order.
e. The values in descending order.

(iii)

In [18]:

```
import pandas as pd
data = pd.Series([4, 7, -5, 3, np.nan])
#Answer to (a)
data_with_index=pd.Series(data.values,index=['d','b','a','c','e'])
print(data_with_index)
#answer to (b)
print(data.min())
#answer to (c)
```

```
print(data.max())
#answer to(d)
print(data.sort_values())
#answer to(e)
print(data.sort_values(ascending=False))
```

```
d    4.0
b    7.0
a   -5.0
c    3.0
e    NaN
dtype: float64
-5.0
7.0
2   -5.0
3    3.0
0    4.0
1    7.0
4    NaN
dtype: float64
1    7.0
0    4.0
3    3.0
2   -5.0
4    NaN
dtype: float64
```

IV(a): Create a 2D array from list of lists and Find the minimum value along each of the three rows

List1=[[110,102,183],[40,175,106],[192,40,195]]

In [20]:

```
import numpy as np
List1=[[110,102,183],[40,175,106],[192,40,195]]
arr=np.array(List1)
minimum_along_row=arr.min(axis=1)
print("Minimum along first row : ",minimum_along_row[0])
print("Minimum along second row : ",minimum_along_row[1])
print("Minimum along third row : ",minimum_along_row[2])
```

```
Minimum along first row :  102
Minimum along second row :  40
Minimum along third row :  40
```

IV(b) Create a numpy array form the given list of lists and Swap row 1 and row 2 in the given array

List1=[[110,102,183],[40,175,106],[192,40,195]]

In [22]:

```
import numpy as np
List1=[[110,102,183],[40,175,106],[192,40,195]]
arr=np.array(List1)
print("Original Array :\n",arr)
arr[0,1]=arr[1,0]
print("Array after swapping :\n",arr)
```

```
Original Array :
 [[110 102 183]
 [ 40 175 106]
 [192  40 195]]
Array after swapping :
 [[110  40 183]
 [ 40 175 106]
 [192  40 195]]
```

IV(c): Create a numpy array form the given list of lists and Replace all the odd numbers in the array with -2

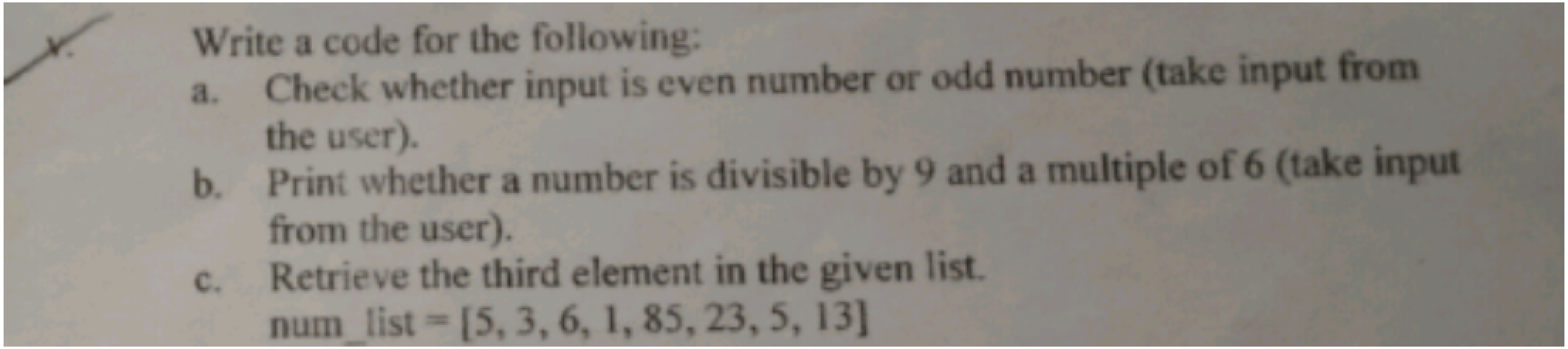
List1=[[110,102,183],[40,175,106],[192,40,195]]

In [23]:

```
#Answer
import numpy as np
List1=[[110,102,183],[40,175,106],[192,40,195]]
arr=np.array(List1)
print(arr)
arr=np.where(arr%2!=0,-2,arr)
print(arr)
```

```
[[110 102 183]
 [ 40 175 106]
 [192  40 195]]
[[110 102  -2]
 [ 40  -2 106]
 [192  40  -2]]
```

(V)



Write a code for the following:

a. Check whether input is even number or odd number (take input from the user).

b. Print whether a number is divisible by 9 and a multiple of 6 (take input from the user).

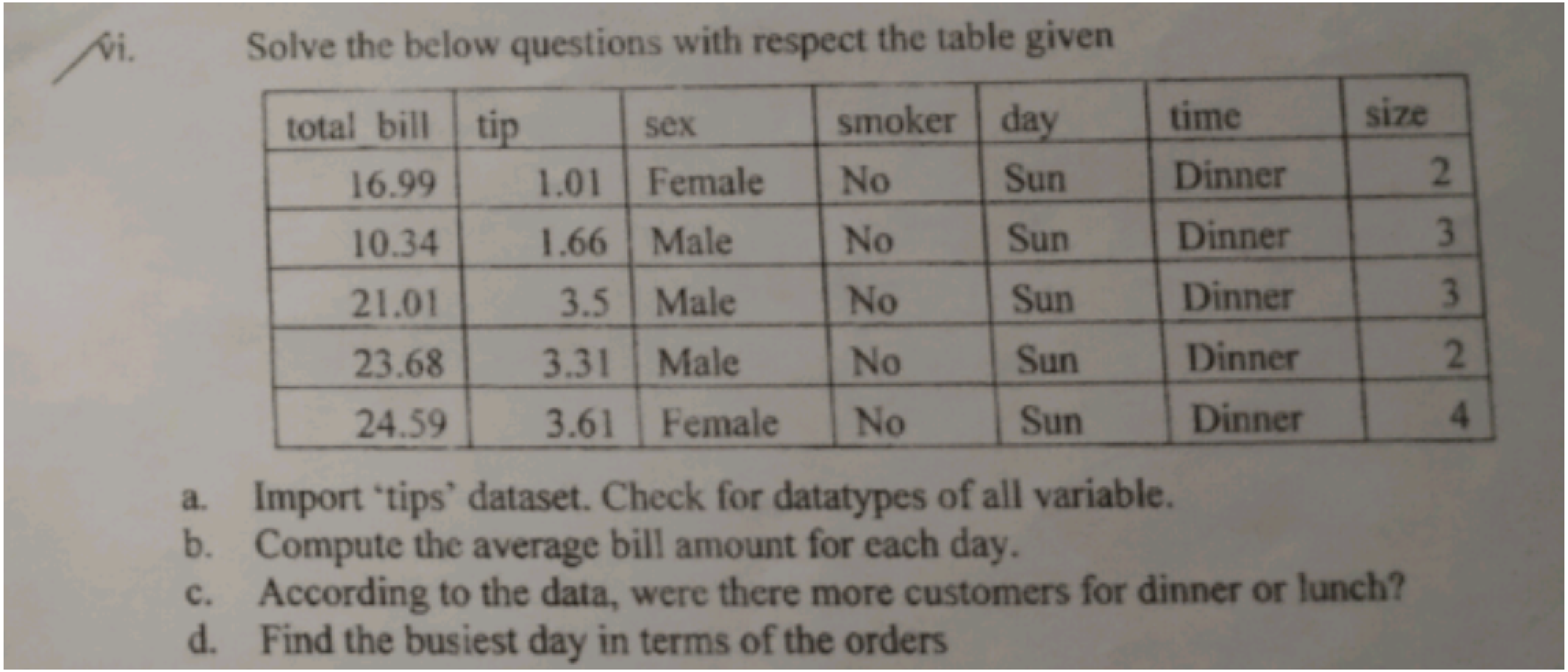
c. Retrieve the third element in the given list.
num_list = [5, 3, 6, 1, 85, 23, 5, 13]

In [29]:

```
#Answer (a)
a=int(input("Enter a number : "))
print('Even' if a%2==0 else 'Odd')
#Answer (b)
a=int(input("Enter a number : "))
print("Yes! It's a divisible by 9 and a multiple of 6" if (a%9==0 and a%6==0) else "No! It isn't a divisible by 9 and a multiple of 6")
#Answer (c)
num_list=[5,3,6,1,85,23,5,13]
print("third element is : " ,num_list[2])
```

```
Enter a number : 75
Odd
Enter a number : 65
No! It isn't a divisible by 9 and a multiple of 6
third element is :  6
```

(VI)



vi. Solve the below questions with respect the table given

total_bill	tip	sex	smoker	day	time	size
16.99	1.01	Female	No	Sun	Dinner	2
10.34	1.66	Male	No	Sun	Dinner	3
21.01	3.5	Male	No	Sun	Dinner	3
23.68	3.31	Male	No	Sun	Dinner	2
24.59	3.61	Female	No	Sun	Dinner	4

a. Import 'tips' dataset. Check for datatypes of all variable.
b. Compute the average bill amount for each day.
c. According to the data, were there more customers for dinner or lunch?
d. Find the busiest day in terms of the orders

The data give is wrong, but we have to assume and do this questions

```
In [ ]:
```