Charotar University of Science and Technology [CHARUSAT]

Chandubhai S. Patel Institute of Technology [CSPIT]

U & P U. Patel Department of Computer Engineering

Assignment Problem

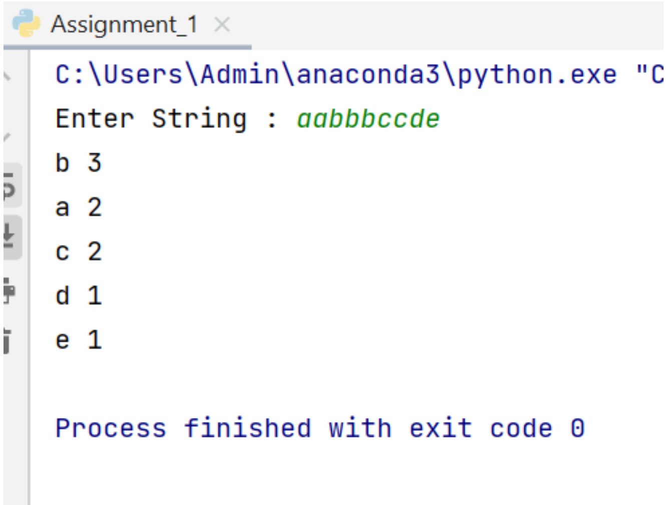
1

```
You are given a string. Your task is to count the frequency of
letters of the string and
print the letters in descending order of frequency.
If the following string is given as input to the program:
aabbbccde
Then, the output of the program should be:
b 3
a 2
c 2
d 1
e 1
```

Code

```python
st = input("Enter String : ")

dicts = {}

for ch in st:
    if ch in dicts:
        dicts[ch] += 1
    else:
        dicts[ch] = 1

l = len(set(st))

for i in range(0, l):
    maxli = [0, 0]
    for key in dicts:
        if dicts[key] > maxli[1]:
            maxli[0] = key
            maxli[1] = dicts[key]
    print(maxli[0] + " " + str(maxli[1]))
    dicts.pop(maxli[0])
```

Output



```
C:\Users\Admin\anaconda3\python.exe "C
Enter String : aabbbccde
b 3
a 2
c 2
d 1
e 1

Process finished with exit code 0
```

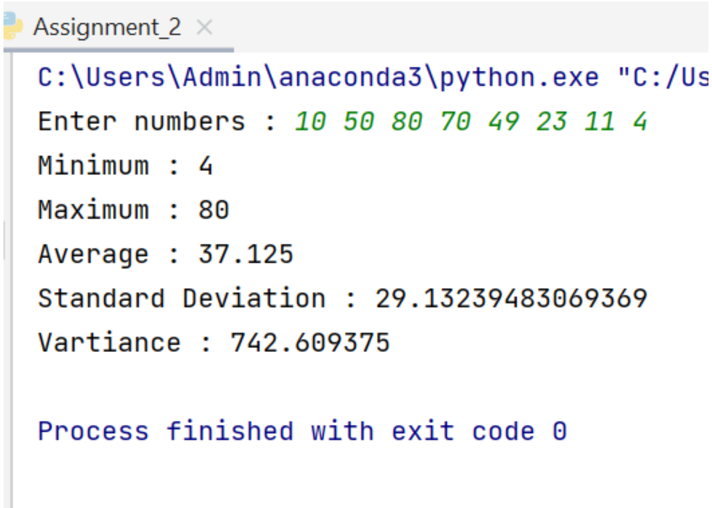
2

Write a procedure to find min, max, mean, standard deviation, variance of number list.
Exampl e
Input : 10 50 80 70 49 23 11 4
output : 4 80 37. 13 27. 25 848. 70

Code

```
numbers = list(map(int,input("Enter numbers : ").split()))
import numpy
import statistics

def Average(lst):
    return sum(lst) / len(lst)

print("Minimum : " + str(min(numbers)))

print("Maximum : " + str(max(numbers)))
print("Average : " + str(Average(numbers)))

print("Standard Deviation : " + str(statistics.stdev(numbers)))

print("Vartiance : " + str(numpy.var(numbers)))
```

Output



```
C:\Users\Admin\anaconda3\python.exe "C:/Us
Enter numbers : 10 50 80 70 49 23 11 4
Minimum : 4
Maximum : 80
Average : 37.125
Standard Deviation : 29.13239483069369
Vartiance : 742.609375

Process finished with exit code 0
```

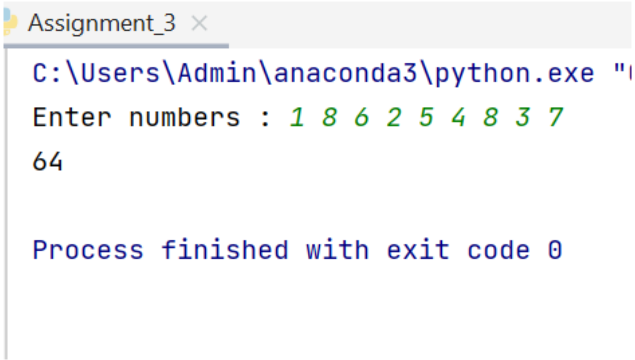
3

You are gi ven an i nteger array hei ght of lengt h n. There are n vertical lines dra wn such
that the t wo endpoi nts of the ith line are (i, 0) and (i, hei ght[i]).
Find two lines that together with the x-axis form a container, such that the container
contains the most water.
Return the maximum amount of water a container can store.
Notice that you may not slant the container.
Input: height = [1, 8, 6, 2, 5, 4, 8, 3, 7]
Output: 49
Explanation: The above vertical lines are represented by array [1, 8, 6, 2, 5,4, 8, 3, 7]. In
this case, the max area of water (blue section) the container can contain is 49.
Example 2:
Input: height = [1, 1]
Output: 1

Code

```
heights = list(map(int,input("Enter numbers : ").split()))

maxh = 0

for i in heights:
    if i > maxh:
        maxh=i


maxh2 = 0
heights.pop(heights.index(maxh))
for i in heights:
    if i > maxh2:
        maxh2=i

print(maxh2*maxh2)
```

Output



```
Assignment_3
C:\Users\Admin\anaconda3\python.exe "
Enter numbers : 1 8 6 2 5 4 8 3 7
64

Process finished with exit code 0
```

4

Given a list of integers, write a program to print the count of all possible unique combinations of numbers whose sum is equal to K.

Input

The first line of input will contain space-separated integers.

The second line of input will contain an integer, denoting K.

Output

The output should be containing the count of all unique combinations of numbers whose sum is equal to K.

Sample Input 1

```
2 4 6 1 3
6
```

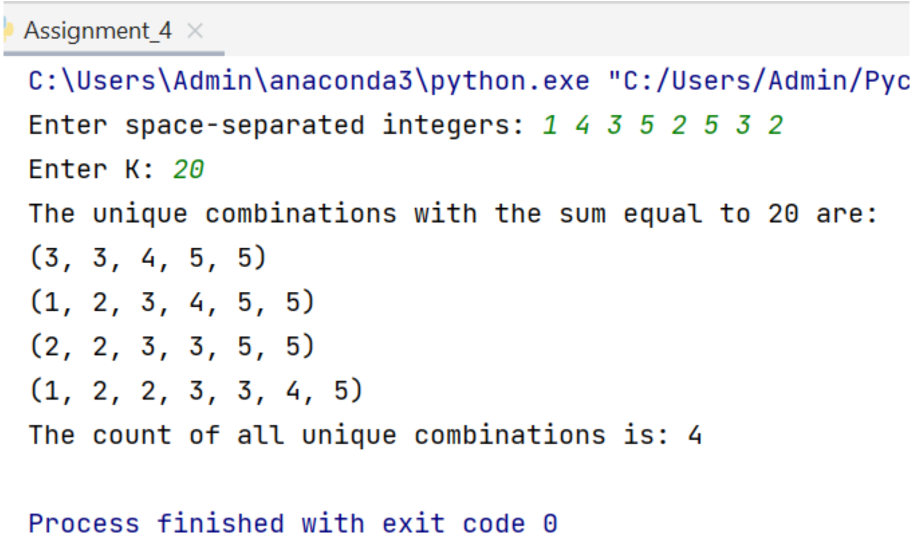
Sample Output 1

```
3
```

Code

```
from itertools import combinations

values =[int(i) for i in input('Enter space-separated integers:
').split()]
values.sort()

K = int(input('Enter K: '))
counterUniqueCombinations=0
print("The unique combinations with the sum equal to "+str(K)+"
are:")
for i in range(1, len(values)+1):
    com =list(set(combinations(values, i)))
    for combination in com:
        if sum(combination) == K:
            print(combination)
            counterUniqueCombinations += 1
print("The count of all unique combinations is:
"+str(counterUniqueCombinations))
```

Output



```
C:\Users\Admin\anaconda3\python.exe "C:/Users/Admin/Pyc
Enter space-separated integers: 1 4 3 5 2 5 3 2
Enter K: 20
The unique combinations with the sum equal to 20 are:
(3, 3, 4, 5, 5)
(1, 2, 3, 4, 5, 5)
(2, 2, 3, 3, 5, 5)
(1, 2, 2, 3, 3, 4, 5)
The count of all unique combinations is: 4

Process finished with exit code 0
```

5

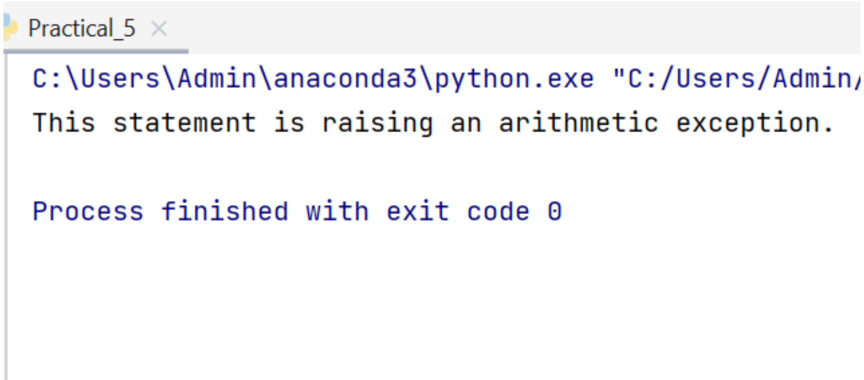
Explain about the different types of Exceptions in Python with suitable example.

Code

Some of the basic inbuilt exceptions are:
1.exception ArithmeticError
This class is the base class for those built-in exceptions that are raised for various arithmetic errors such as:

- OverflowError
- ZeroDivisionError
- FloatingPointError

```
  try:
    a = 10/0
    print (a)
except ArithmeticError:
        print ("This statement is raising an arithmetic exception.")
else:
    print ("Success.")
```

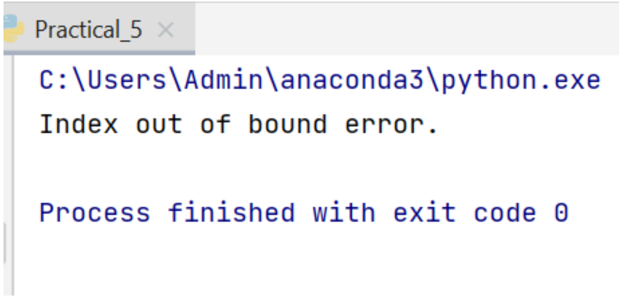


```
C:\Users\Admin\anaconda3\python.exe "C:/Users/Admin/
This statement is raising an arithmetic exception.

Process finished with exit code 0
```

2.exception LookupError
This is the base class for those exceptions that are raised when a key or index used on mapping or sequence is invalid or not found. The exceptions raised are :

- KeyError
- IndexError

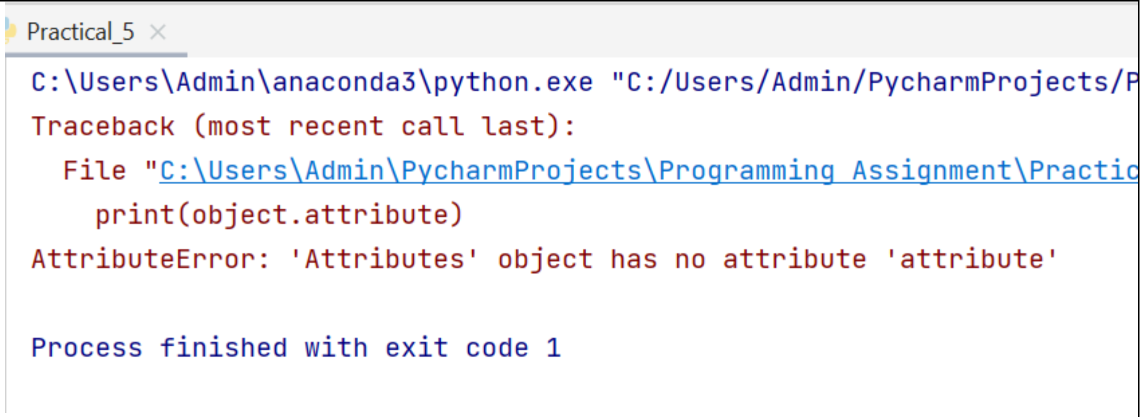
```
try:
    a = [1, 2, 3]
    print (a[3])
except LookupError:
    print ("Index out of bound error.")
```



```
C:\Users\Admin\anaconda3\python.exe
Index out of bound error.

Process finished with exit code 0
```

3.exception AttributeError
An AttributeError is raised when an attribute reference or assignment fails such as when a non-existent attribute is referenced.

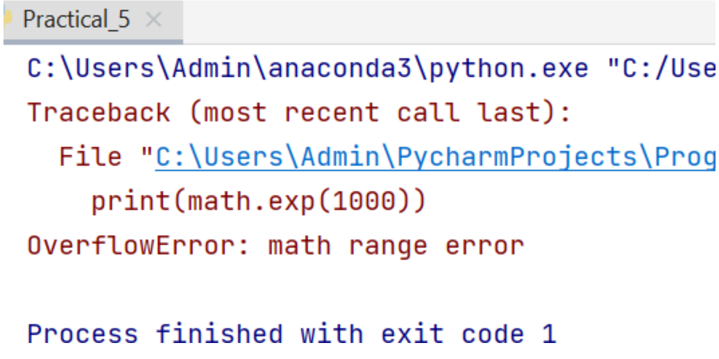
```
class Attributes(object):
    pass
```



```
C:\Users\Admin\anaconda3\python.exe "C:/Users/Admin/PycharmProjects/P
Traceback (most recent call last):
  File "C:\Users\Admin\PycharmProjects\Programming Assignment\Practic
    print(object.attribute)
AttributeError: 'Attributes' object has no attribute 'attribute'

Process finished with exit code 1
```

4.exception FloatingPointError
A FloatingPointError is raised when a floating point operation fails. This exception is always defined, but can only be raised when Python is configured with the-with-fpectl option, or the WANT_SIGFPE_HANDLER symbol is defined in the pyconfig.h file.

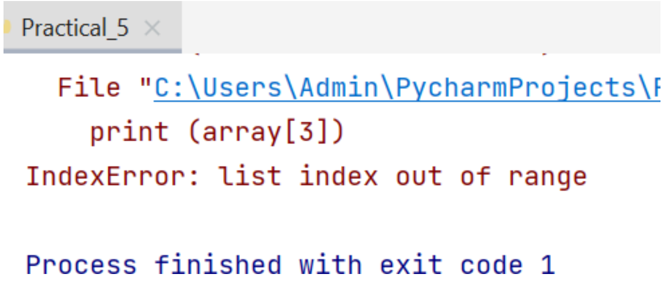
```
import math

print(math.exp(1000))
```



```
C:\Users\Admin\anaconda3\python.exe "C:/Use
Traceback (most recent call last):
  File "C:\Users\Admin\PycharmProjects\Prog
    print(math.exp(1000))
OverflowError: math range error

Process finished with exit code 1
```

5.exception IndexError
An IndexError is raised when a sequence is referenced which is out of range.

```
array = [ 0, 1, 2 ]
print (array[3])
```



```
  File "C:\Users\Admin\PycharmProjects\F
    print (array[3])
IndexError: list index out of range

Process finished with exit code 1
```

7 Write a django code to send an email with attachment

Code

```
from django.shortcuts import render
from .forms import ContactForm
from django.core.mail import send_mail

def contactview(request):
    name=''
    email=''
    comment=''


    form= ContactForm(request.POST or None)
    if form.is_valid():
        name= form.cleaned_data.get("name")
        email= form.cleaned_data.get("email")
        comment=form.cleaned_data.get("comment")


        comment= name + " with the email, " + email + ", sent the
following message:\n\n" + comment;
        send_mail('The title of this post', comment,
'admin@gmail.com', ['admin@gmail.com'])


        context= {'form': form}

        return render(request, 'contact/contact.html', context)

    else:
        context= {'form': form}
        return render(request, 'contact/contact.html', context)
```

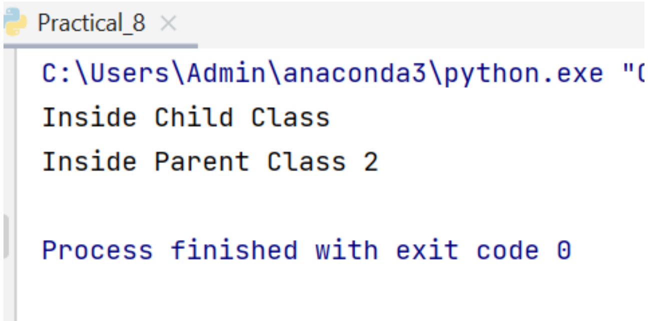
8 Program to demonstrate the Overriding of the Base Class method in the Derived Class

Code

```
class P1_class():

    def show(self):
        print("Inside Parent Class 1")


class P2_class():

    def display(self):
        print("Inside Parent Class 2")
```

```python
class Child_class(P1_class, P2_class):

    def show(self):
        print("Inside Child Class")


obj = Child_class()

obj.show()
obj.display()
```

Output



```
C:\Users\Admin\anaconda3\python.exe "(
Inside Child Class
Inside Parent Class 2

Process finished with exit code 0
```

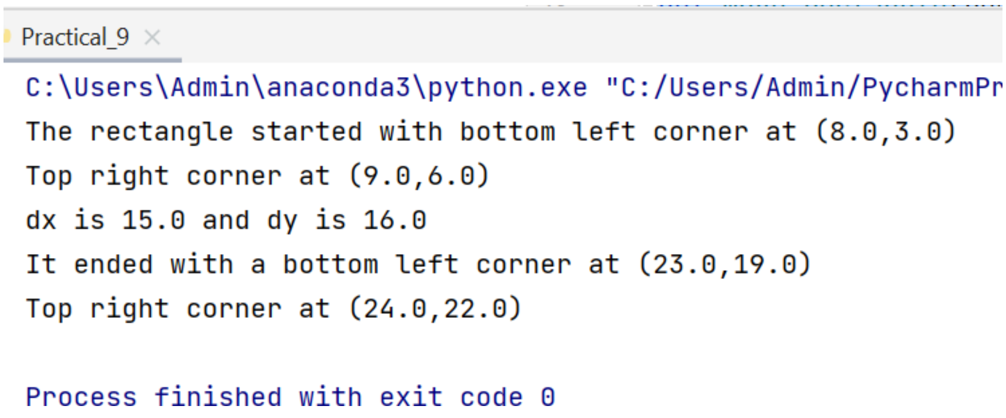
9

Write Pythonic code to create a function named move_rectangle() that takes an object of Rectangle class and two numbers named dx and dy. It should change the location of the Rectangle by adding dx to the x coordinate of corner and adding dy to the y coordinate of corner.

Code

```python
class Point(object):
    pass

class Rectangle(object):
    pass

rectangle = Rectangle()

bottom_left = Point()
bottom_left.x = 8.0
bottom_left.y = 3.0

top_right = Point()
top_right.x = 9.0
top_right.y = 6.0

rectangle.corner1 = bottom_left
rectangle.corner2 = top_right

dx = 15.0
dy = 16.0

def move_rectangle(rectangle, dx, dy):
    print(f"The rectangle started with bottom left corner at
({rectangle.corner1.x},{rectangle.corner1.y})"
          f"\nTop right corner at
({rectangle.corner2.x},{rectangle.corner2.y})"
          f"\ndx is {dx} and dy is {dy}")
    rectangle.corner1.x = rectangle.corner1.x + dx
    rectangle.corner2.x = rectangle.corner2.x + dx
    rectangle.corner1.y = rectangle.corner1.y + dy
    rectangle.corner2.y = rectangle.corner2.y + dy
    print(f"It ended with a bottom left corner at
({rectangle.corner1.x},{rectangle.corner1.y})"
          f"\nTop right corner at
({rectangle.corner2.x},{rectangle.corner2.y})")

move_rectangle(rectangle, dx, dy)
```

Output



```
C:\Users\Admin\anaconda3\python.exe "C:/Users/Admin/PycharmPr
The rectangle started with bottom left corner at (8.0,3.0)
Top right corner at (9.0,6.0)
dx is 15.0 and dy is 16.0
It ended with a bottom left corner at (23.0,19.0)
Top right corner at (24.0,22.0)

Process finished with exit code 0
```