College of Technology and Engineering, MPUAT, Udaipur

Name – Abhishek Pandey Class – B.Tech III yr

Subject – Data Analysis with Python (CS- 366) Semester – VI

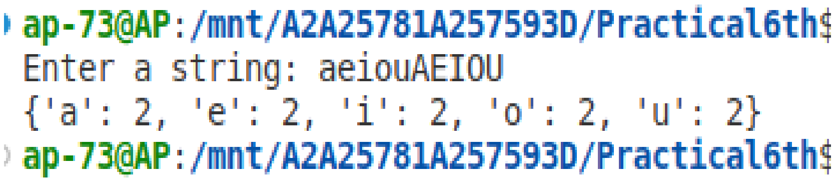
Program – 35

Object – Write a program to count the number of each vowel.

Code :

```
def count_vowels(string):
    vowels = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0}
    for char in string:
        if char.lower() in vowels:
            vowels[char.lower()] += 1
    return vowels

string = input("Enter a string: ")
vowel_count = count_vowels(string)
print(vowel_count)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
Enter a string: aeiouAEIOU
{'a': 2, 'e': 2, 'i': 2, 'o': 2, 'u': 2}
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```

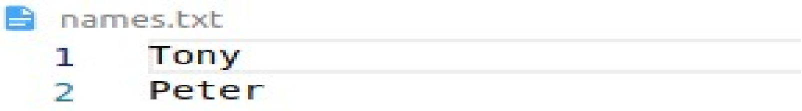
Program – 36

Object – Write a prgram to mail merge.

Code :

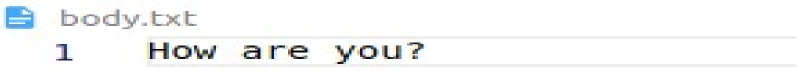
```
# names are in the names.txt
# body of the mail is in body.txt

#open names.txt for reading
with open("names.txt", 'r', encoding='utf-8') as names_file:

    #open body.txt for reading
    with open("body.txt", 'r', encoding='utf-8') as body_file:

        #read the entire content of body file
        body = body_file.read()

        #iterates over names
        for name in names_file:
            mail = "Hello " + name.strip() + "\n" + body

            #write the mails to individual file
            with open(name.strip() + ".txt", 'w', encoding='utf-8') as mail_file:
                mail_file.write(mail)
```

Output :



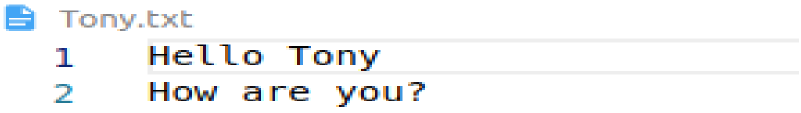
names.txt

```
Tony
Peter
```



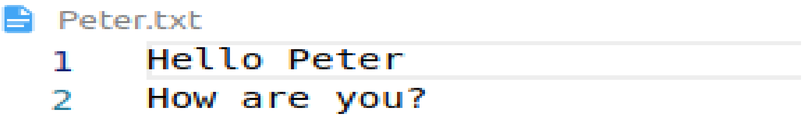
body.txt

```
How are you?
```



Tony.txt

```
Hello Tony
How are you?
```



Peter.txt

```
Hello Peter
How are you?
```

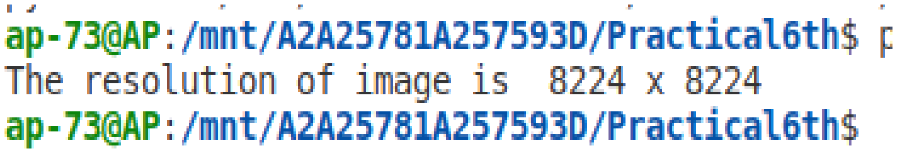
Program – 37

Object – Write a program to find the size (resolution) of image.

Code :

```
def image_res(filename):
    #this function pprints resolution of image passed to it
    #open image for reading in binary mode
    with open(filename, 'rb') as img_file:

        #height of image (in 2 bytes) is at 164th position
        img_file.seek(1050)

        #read the 2 bytes
        a = img_file.read(2)

        #calculate the height
        height = (a[0] << 8) + a[1]

        #next 2 bytes is width
        a = img_file.read(2)

        #calculate width
        width = (a[0] << 8) + a[1]

    print("The resolution of image is ", width, "x", height)

#main function
image_res("flower.jpg")
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ p
The resolution of image is  8224 x 8224
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```

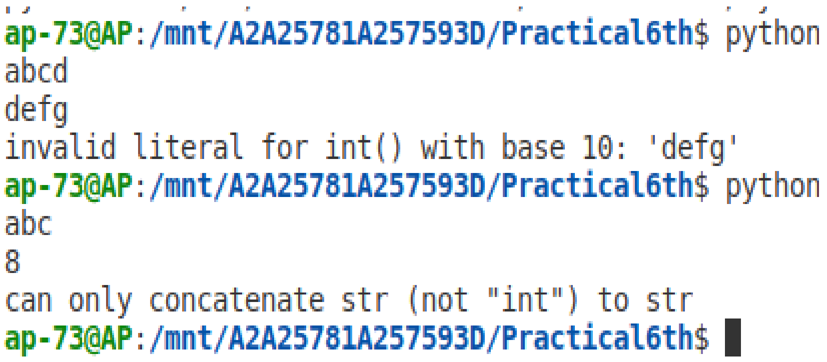
Program – 38

Object – Write a program to catch multiple excptions as a paranthesized tuple (in one line).

Code :

```
string = input()
try:
    num = int(input())
    print(string+num)

except(TypeError, ValueError) as e:
    print(e)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
abcd
defg
invalid literal for int() with base 10: 'defg'
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
abc
8
can only concatenate str (not "int") to str
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ 
```

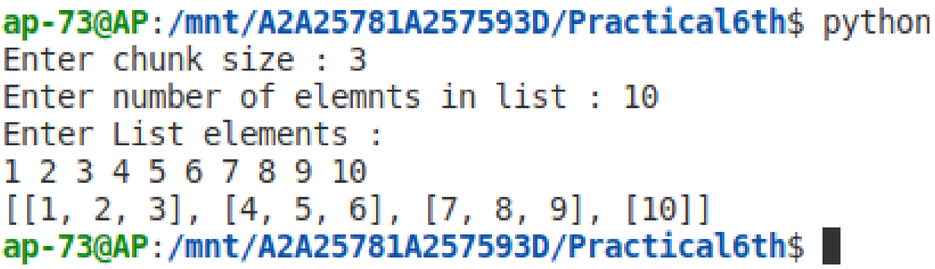
Program – 39

Object – Write a program split a list into evenly sized chunks.

Code :

```
def chunkify(lst, chunk_size):
        return [lst[i:i+chunk_size] for i in range(0, len(lst), chunk_size)]
#ask for chunk size from user
chunk_size = int(input("Enter chunk size : "))

my_list = []
size = int(input("Enter number of elemnts in list : "))

print("Enter List elements : ")
input_str = input()
# split the input string into a list of strings
input_list = input_str.split()

for i in range(size):
    element = int(input_list[i])
    my_list.append(element)

chunked_list = chunkify(my_list, chunk_size)
print(chunked_list)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
Enter chunk size : 3
Enter number of elemnts in list : 10
Enter List elements :
1 2 3 4 5 6 7 8 9 10
[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10]]
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```

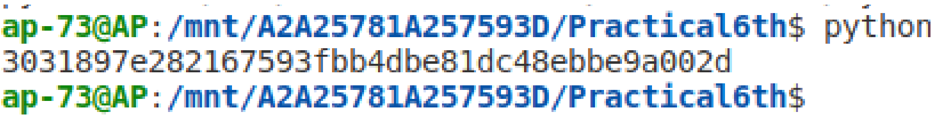
Program – 40

Object – Write a program to find the hash of a file & display it.

Code :

```
import hashlib
def hashFile(filename):
    # this function return the SHA1 hash of the file passed into it
    #make a hash object
    hash_object = hashlib.sha1()

    #open file for reading in binary mode
    with open(filename, 'rb') as file:
        #loop till end of file
        while True:
            chunk = file.read(1024)
            if not chunk:
                break
            hash_object.update(chunk)

    return hash_object.hexdigest()

#main function
message = hashFile("body.txt")
print(message)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
3031897e282167593fbb4dbe81dc48ebbe9a002d
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```

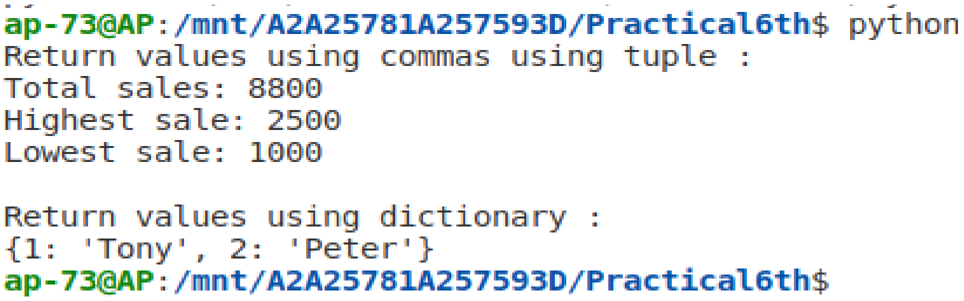
Program – 41

Object – Write a program to return multiple values from a function.

Code :

```
def calculate_sales_info(sales_list):
    # Calculate the total sales, highest sale, and lowest sale
    total_sales = sum(sales_list)
    highest_sale = max(sales_list)
    lowest_sale = min(sales_list)
    # Return the results as a tuple
    return total_sales, highest_sale, lowest_sale

#return values using dictionary
def name():
    n1 = "Tony"
    n2 = "Peter"
    return {1:n1, 2:n2}

# Call the function with a list of sales figures
sales_figures = [1000, 2000, 1500, 2500, 1800]
total_sales, highest_sale, lowest_sale = calculate_sales_info(sales_figures)

# Print the results
print("Return values using commas using tuple : ")
print("Total sales:", total_sales)
print("Highest sale:", highest_sale)
print("Lowest sale:", lowest_sale)
print("\nReturn values using dictionary : ")
names = name()
print(names)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
Return values using commas using tuple :
Total sales: 8800
Highest sale: 2500
Lowest sale: 1000

Return values using dictionary :
{1: 'Tony', 2: 'Peter'}
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```

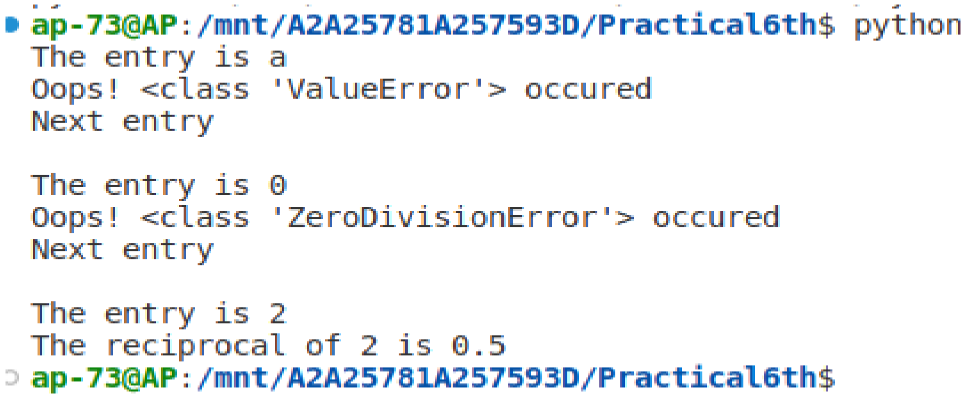
Program – 42

Object – Write a program for catching exceptions in python.

Code :

```
import sys

random_list = ['a', 0, 2]

for entry in random_list:
    try:
        print("The entry is",entry)
        res = 1/int(entry)
        break
    except:
        print("Oops!", sys.exc_info()[0],"occured")
        print("Next entry\n")
print("The reciprocal of", entry, "is", res)
```

Output :



```
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$ python
The entry is a
Oops! <class 'ValueError'> occured
Next entry

The entry is 0
Oops! <class 'ZeroDivisionError'> occured
Next entry

The entry is 2
The reciprocal of 2 is 0.5
ap-73@AP:/mnt/A2A25781A257593D/Practical6th$
```