PYTHON PROGRAM-4

1.Write a python program to display the keys alphabetically using dictionary

For example:

Result
Keys are 1 2 3 4 5 6

PROGRAM:

```
a={6,5,4,3,2,1}
b=sorted(a)
print("Keys are\n",end="")
print(" ".join(map(str,b)))
```

OUTPUT:

	Expected	Got	
✔	Keys are 1 2 3 4 5 6	Keys are 1 2 3 4 5 6	✔

Passed all tests! ✔

2.Place result="You can't divide with 0" to the right place so that program avoids ZeroDivisionError.

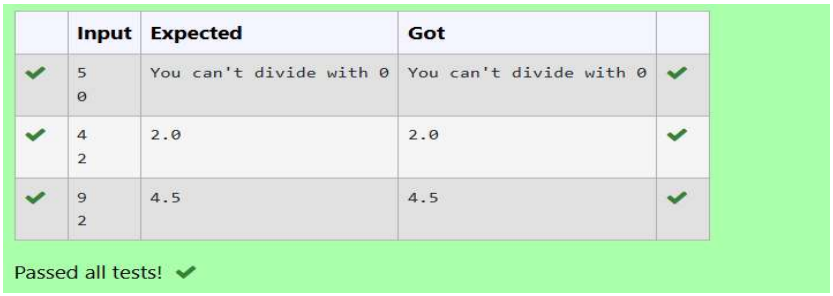
For example:

Input	Result
5 0	You can't divide with 0

PROGRAM:

```
a=int(input())
b=int(input())
try:
   result=a/b
   print(result)
except ZeroDivisionError:
   print("You can't divide with 0")
```

OUTPUT:



	Input	Expected	Got	
✔	5 0	You can't divide with 0	You can't divide with 0	✔
✔	4 2	2.0	2.0	✔
✔	9 2	4.5	4.5	✔

Passed all tests! ✔

3.[Facebook] Write a Python program to merge two files into a third file.

For example:

Test	Result

`file1_path = 'test_case_2_file1.txt'`
`file2_path = 'test_case_2_file2.txt'`
`output_file_path = 'test_case_2_merged.txt'`
`create_file(file1_path, "Line 1 of file 1\nLine 2 of file 1\n")`
`create_file(file2_path, "")`
`merge_files(file1_path, file2_path, output_file_path)`
`print(read_file(output_file_path))`
	`Line 1 of file 1`
	`Line 2 of file 1`

PROGRAM:

```
def create_file(file_path, content):
    with open(file_path, 'w') as file:
        file.write(content)
def merge_files(file1_path, file2_path, output_file_path):
    with open(file1_path,'r') as file1:
        d=file1.read()
    with open(file2_path,'r') as file2:
        s=file2.read()
    with open(output_file_path,'w') as file:
        file.write(d+s)
def read_file(file_path):
    with open(file_path, 'r') as file:
        return file.read()
```

OUTPUT:

	Test	Expected	Got	
✔	file1_path = 'test_case_1_file1.txt' file2_path = 'test_case_1_file2.txt' output_file_path = 'test_case_1_merged.txt' create_file(file1_path, "Line 1 of file 1\nLine 2 of file 1\n") create_file(file2_path, "Line 1 of file 2\nLine 2 of file 2\n") merge_files(file1_path, file2_path, output_file_path) print(read_file(output_file_path))	Line 1 of file 1 Line 2 of file 1 Line 1 of file 2 Line 2 of file 2	Line 1 of file 1 Line 2 of file 1 Line 1 of file 2 Line 2 of file 2	✔
✔	file1_path = 'test_case_2_file1.txt' file2_path = 'test_case_2_file2.txt' output_file_path = 'test_case_2_merged.txt' create_file(file1_path, "Line 1 of file 1\nLine 2 of file 1\n") create_file(file2_path, "") merge_files(file1_path, file2_path, output_file_path) print(read_file(output_file_path))	Line 1 of file 1 Line 2 of file 1	Line 1 of file 1 Line 2 of file 1	✔

Passed all tests! ✔

4.Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case.

For example:

Input	Result
saveetha	SAVEETHA

PROGRAM:

```
class string:
    def __init__(self):
        self.s=""
    def get_string(self,s):
        self.s=s
    def print_string(self):
        print(self.s.upper())
s=str(input())
muth=string()
muth.get_string(s)
muth.print_string()
```

OUTPUT:

	Input	Expected	Got	
✔	saveetha	SAVEETHA	SAVEETHA	✔
✔	engineering	ENGINEERING	ENGINEERING	✔
✔	college	COLLEGE	COLLEGE	✔

Passed all tests! ✔

5.Write a python program to print the size of dictionary using getsizeof() from sys module

dic1 = {"A": 1, "B": 2, "C": 3}

dic2 = {"Geek1": "Raju", "Geek2": "Nikhil", "Geek3": "Deepanshu"}

dic3 = {1: "Lion", 2: "Tiger", 3: "Fox", 4: "Wolf"}

For example:

Result
Size of dic1: 232 bytes Size of dic2: 232 bytes Size of dic3: 232 bytes

PROGRAM:

```
import sys
dic1 = {"A": 1, "B": 2, "C": 3}
dic2= {"Geek1": "Raju", "Geek2": "Nikhil", "Geek3":
"Deepanshu"}
dic3={1: "Lion", 2: "Tiger", 3: "Fox", 4: "Wolf"}
print(f"Size of dic1: {sys.getsizeof(dic1)}bytes")
print(f"Size of dic2: {sys.getsizeof(dic2)}bytes")
print(f"Size of dic3: {sys.getsizeof(dic3)}bytes")
```

OUTPUT:

	Expected	Got	
✔	Size of dic1: 232bytes Size of dic2: 232bytes Size of dic3: 232bytes	Size of dic1: 232bytes Size of dic2: 232bytes Size of dic3: 232bytes	✔

Passed all tests! ✔