Coding and solution

Team Id	PNT2022TMID41774
Project name	Emerging methods for early detection of forest fire
Maximum marks	2 Marks

Debugging And Traceability

* Debugging is to do a performance analysis, which can't be accomplished on the debugging end.

* 5 staghe main advantage of using trace over des of debugging

ΔDenial

ΔAnger

ΔBargaining

ΔDepression

Δ Acceptance

* Then we have the pdb prompt. Now to navigate the code, we can use the following commands.

* Post-mortem debugging means entering debug mode after the program is finished with the execution process (failure has already occurred).

* supports post-mortem debugging through the pm() and post_mortem() functions.

Debugging python

Python3

```
a = 20
b = 10

s = 0
for i in range(a):
    s += a / b
    b -= 1
```

Output:

```
C:\Users\admin\Desktop\CS_Stuff\pythoVS>python exppdb.py
> c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py(10)<module>()
-> x = input("Enter first number : ")
(Pdb)
```

Traceability in python

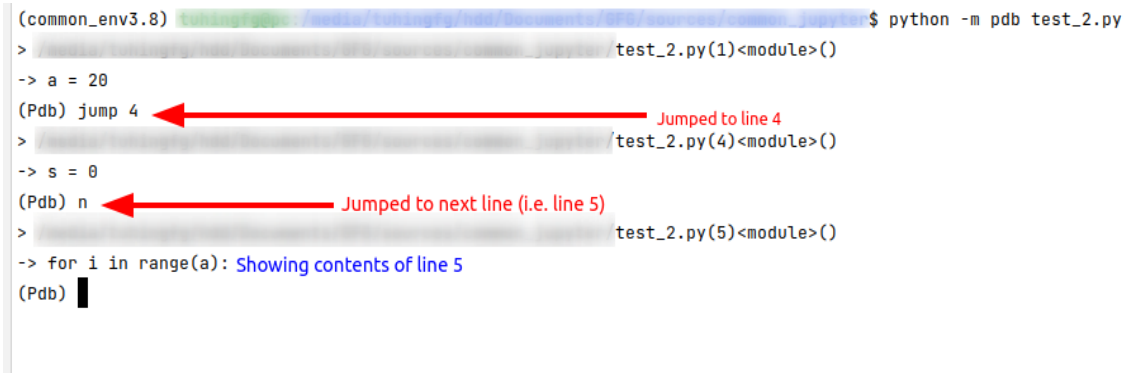
Python3

```python
import pdb

def addition(a, b):
    answer = a * b
    return answer

pdb.set_trace()
x = input("Enter first number : ")
y = input("Enter second number : "
sum = addition(x, y)
print(sum)
```

Output:



Checking variables on the Stack

* All the variables including variables local to the function being executed in the program as well as global are maintained on the stack

* We can use args(or use a) to print all the arguments of a function which is currently active.

* p command evaluates an expression given as an argument and prints the result.

* Python pdb Breakpoints

Δ While working with large programs, we often want to add a number of breakpoints where we know errors might occur.

Δ To do this you just have to use the break command.

Δ When you insert a breakpoint, the debugger assigns a number to it starting from .

Δ Use the break to display all the breakpoints.

Syntax:

break filename: lineno, condition

```
C:\Users\admin\Desktop\CS_Stuff\pythoVS>python -m pdb exppdb.py
> c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py(1)<module>()
-> def multiply(a, b):
(Pdb) break exppdb.py:6
Breakpoint 1 at c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py:6
(Pdb) break exppdb.py:7
Breakpoint 2 at c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py:7
(Pdb) break
Num Type         Disp Enb   Where
1   breakpoint   keep yes   at c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py:6
2   breakpoint   keep yes   at c:\users\admin\desktop\cs_stuff\pythovs\exppdb.py:7
(Pdb)
```