Started on	Thursday, 4 September 2025, 11:37 AM
State	Finished
Completed on	Thursday, 4 September 2025, 11:56 AM
Time taken	18 mins 54 secs
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program using Tuples to accept individual address details and display the Same

Address:

"Saveetha","Thandalam","7485961230"

For example:

Result
`Name: Saveetha Address: Thandalam phone_no: 7485961230`

Answer: (penalty regime: 0 %)

```
tup=("Saveetha","Thandalam","7485961230")
print("Name: ",tup[0],"Address: ",tup[1],"phone_no: ",tup[2])
```

	Expected	Got	
✔	`Name: Saveetha Address: Thandalam phone_no: 7485961230`	`Name: Saveetha Address: Thandalam phone_no: 7485961230`	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

write a python program to perform addition and subtraction operation using class and if,elif...

note:

class name should be calculator, function name should be setvalues(to set a and b values) add and sub

cases : choice 1 ->perform addition ,choice 2-> perform subtraction , choice 0 -> exiting, other choices -> print 'invalid choice'

For example:

Input	Result
5 5 1 0	Result: 10 Exiting!

Answer: (penalty regime: 0 %)

```
class cse:
    def vinay(self,a,b,c):
        if c==1:
            print("Result: ",a+b)
        if c==2:
            print("Result: ",a-b)
        if c==0:
            print("Exiting!")
a=int(input())
b=int(input())
p=cse()
while True:
    c=int(input())
    p.vinay(a,b,c)
    if c==0:
        break
```

	Input	Expected	Got	
✔	5 5 1 0	Result: 10 Exiting!	Result: 10 Exiting!	✔
✔	5 5 2 0	Result: 0 Exiting!	Result: 0 Exiting!	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Incorrect

Mark 20.00 out of 20.00

1. Create a Python class called **BankAccount** which represents a bank account, having as attributes: **accountNumber** (numeric type), **name** (name of the account owner as string type), balance.
2. Create a **setvalues()** with parameters: **accountNumber, name, balance**.
3. Create a **Deposit()** method which manages the deposit actions.
4. Create a **Withdrawal() method** which manages withdrawals actions.
5. Create an **bankFees()** method to apply the bank fees with a percentage of 5% of the balance account.
6. Create a **display()** method to display account details.
7. Give the complete code for the **BankAccount class**.

For example:

Input	Result
21456398 saveetha 25000	Account Number : 21456398 Account Name : saveetha Account Balance : 24900 $

Answer: (penalty regime: 0 %)

```
class BankAccount:
    # def __init__(self,accountNumber, name, balance):
    #     self.accountNumber=accountNumber
    #     self.name=Name
    #     self.balance=balance
    def display(self,accountNumber,name,balance):
        print("Account Number : ",accountNumber)
        print("Account Name : ",name)
        print("Account Balance : ",balance-100*," $")
accountNumber=int(input())
name=input()
balance=int(input())
p=BankAccount()
p.display(accountNumber,name,balance)
```

Syntax Error(s)

```
  File "__tester__.python3", line 9
    print("Account Balance : ",balance-100*," $")
                                               ^

SyntaxError: invalid syntax
```

Incorrect

Marks for this submission: 0.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

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

Answer: (penalty regime: 0 %)

Reset answer

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

	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! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program that asks the user to enter an integer n and return a dictionary whose keys are integers 1, 2, 3, ... n and whose values are 1! , 2! , 3! , ... , n!

For example:

Input	Result
6	The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120}

Answer: (penalty regime: 0 %)

```
import math
d={}
n=int(input())
for i in range(1,n):
    d[i]=math.factorial(i)
print("The obtained dictionary is d = ",d)

```

	Input	Expected	Got	
✔	6	The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120}	The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120}	✔
✔	4	The obtained dictionary is d = {1: 1, 2: 2, 3: 6}	The obtained dictionary is d = {1: 1, 2: 2, 3: 6}	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.