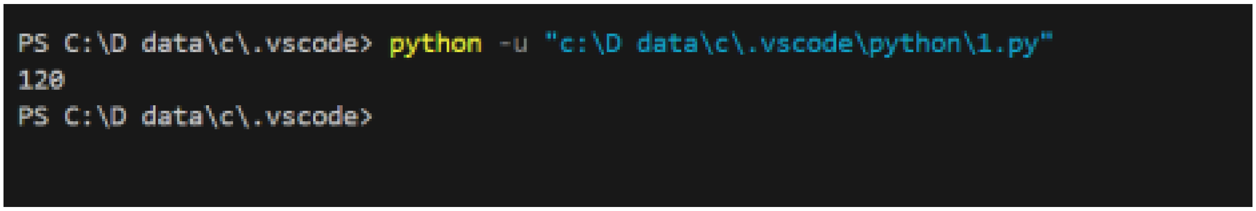
Python assignment 7:

Name:Revanth

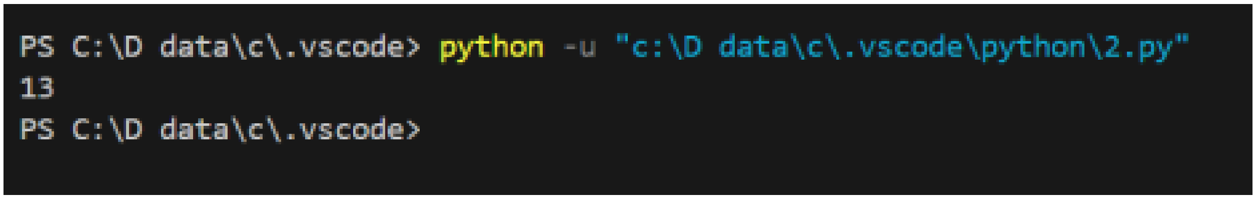
Mis:112315133

```
1.def memorize_factorial():
    cache = {}

    def factorial(n):
        if n in cache:
            return cache[n]
        else:
            if n == 0 or n == 1:
                result = 1
            else:
                result = n * factorial(n-1)
            cache[n] = result
            return result

    return factorial

fact = memorize_factorial()

print(fact(5))
```



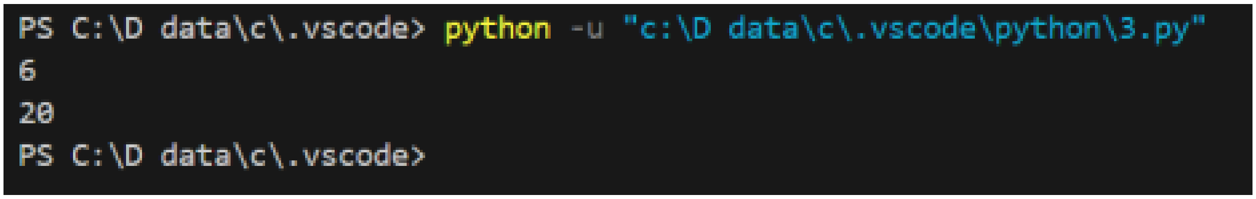
```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\1.py"
120
PS C:\D data\c\.vscode>
```

```
2.def create_pipeline(functions):
    def pipeline(x):
        for f in functions:
            x = f(x)
        return x
    return pipeline
def multiply_by_2(x):
    return x*2
def add_3(x):
    return x+3

pipeline=create_pipeline([multiply_by_2,add_3])
print(pipeline(5))
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\2.py"
13
PS C:\D data\c\.vscode>
```

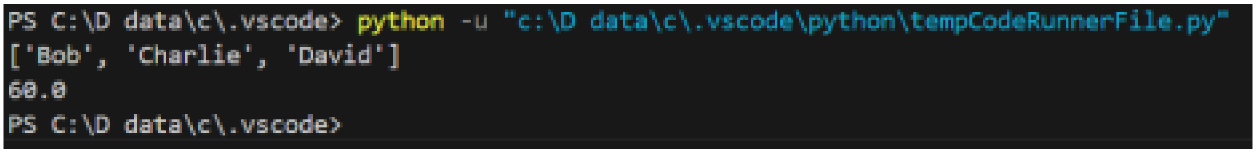
```
3.def multiply(a):
    def inner(b):
        return a * b
    return inner

print(multiply(2)(3))
print(multiply(4)(5))
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\3.py"
6
20
PS C:\D data\c\.vscode>
```

4.

```
from functools import reduce

students = [
    {'name': 'Alice', 'score': 45},
    {'name': 'Bob', 'score': 55},
    {'name': 'Charlie', 'score': 65},
    {'name': 'David', 'score': 75}
]

passed_students = filter(lambda student: student['score'] >= 50, students)
passed_student_names = list(map(lambda student: student['name'],
passed_students))
total_score = reduce(lambda acc, student: acc + student['score'],
students, 0)
average_score = total_score / len(students)
print(passed_student_names)
print(average_score)
```

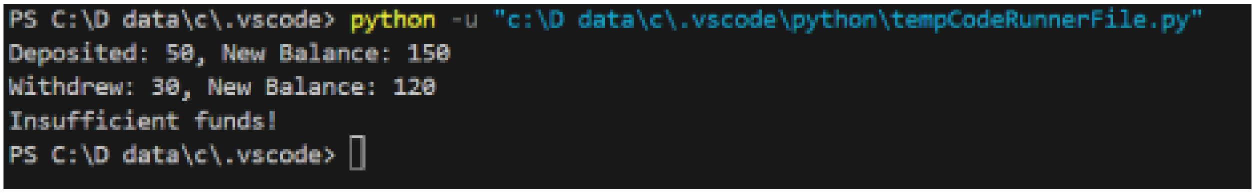


```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
['Bob', 'Charlie', 'David']
60.0
PS C:\D data\c\.vscode>
```

5.

```
def bank_account(initial_balance):
    balance = initial_balance
```

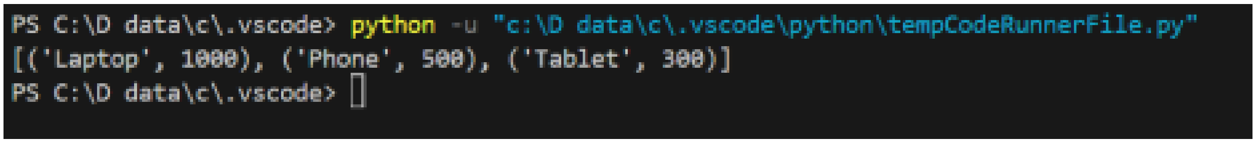
```python
    def deposit(amount):
        nonlocal balance
        balance += amount
        print(f"Deposited: {amount}, New Balance: {balance}")

    def withdraw(amount):
        nonlocal balance
        if amount > balance:
            print("Insufficient funds!")
        else:
            balance -= amount
            print(f"Withdrew: {amount}, New Balance: {balance}")

    return deposit, withdraw


account_deposit, account_withdraw = bank_account(100)
account_deposit(50)
account_withdraw(30)
account_withdraw(200)
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
Deposited: 50, New Balance: 150
Withdrew: 30, New Balance: 120
Insufficient funds!
PS C:\D data\c\.vscode> 
```

6.

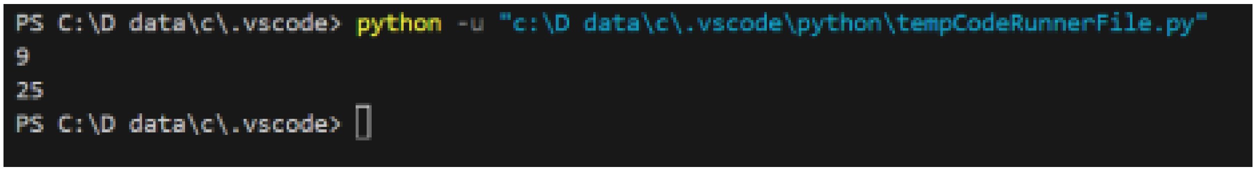
```python
products = [('Laptop', 1000), ('Phone', 500), ('Tablet', 300)]

sorted_products = sorted(products, key=lambda x: x[1],

reverse=True)

print(sorted_products)
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
[('Laptop', 1000), ('Phone', 500), ('Tablet', 300)]
PS C:\D data\c\.vscode> 
```

7.

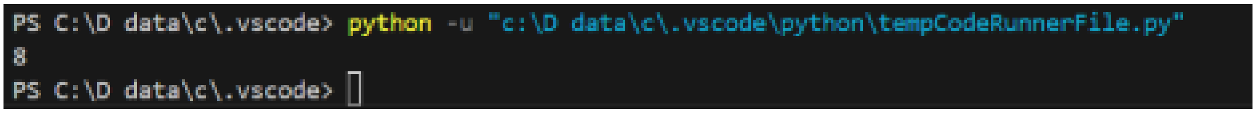
```python
from functools import partial
power_of_two = partial(pow, exp=2)

print(power_of_two(3))
print(power_of_two(5))
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
9
25
PS C:\D data\c\.vscode> 
```

8.

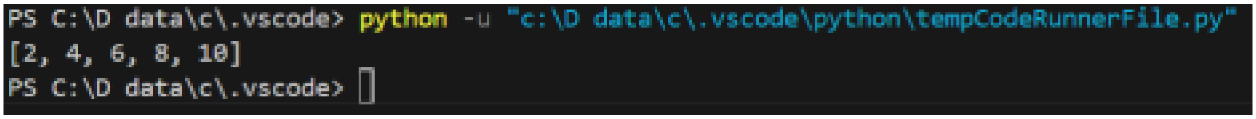
```
def polynomial_creator(*coefficients):
    def polynomial(x):
        result = 0
        for power, coeff in enumerate(reversed(coefficients)):
            result += coeff * (x ** power)
        return result
    return polynomial

p = polynomial_creator(3, 0, -4)
print(p(2))
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
8
PS C:\D data\c\.vscode> 
```

9.

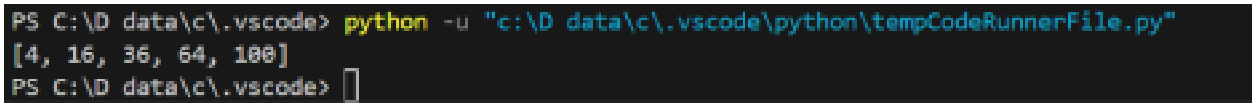
```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
print(even_numbers)
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
[2, 4, 6, 8, 10]
PS C:\D data\c\.vscode> 
```

10.

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
squared_even_numbers = list(map(lambda x: x ** 2, filter(lambda x: x % 2
== 0, numbers)))
print(squared_even_numbers)
```



```
PS C:\D data\c\.vscode> python -u "c:\D data\c\.vscode\python\tempCodeRunnerFile.py"
[4, 16, 36, 64, 100]
PS C:\D data\c\.vscode> 
```