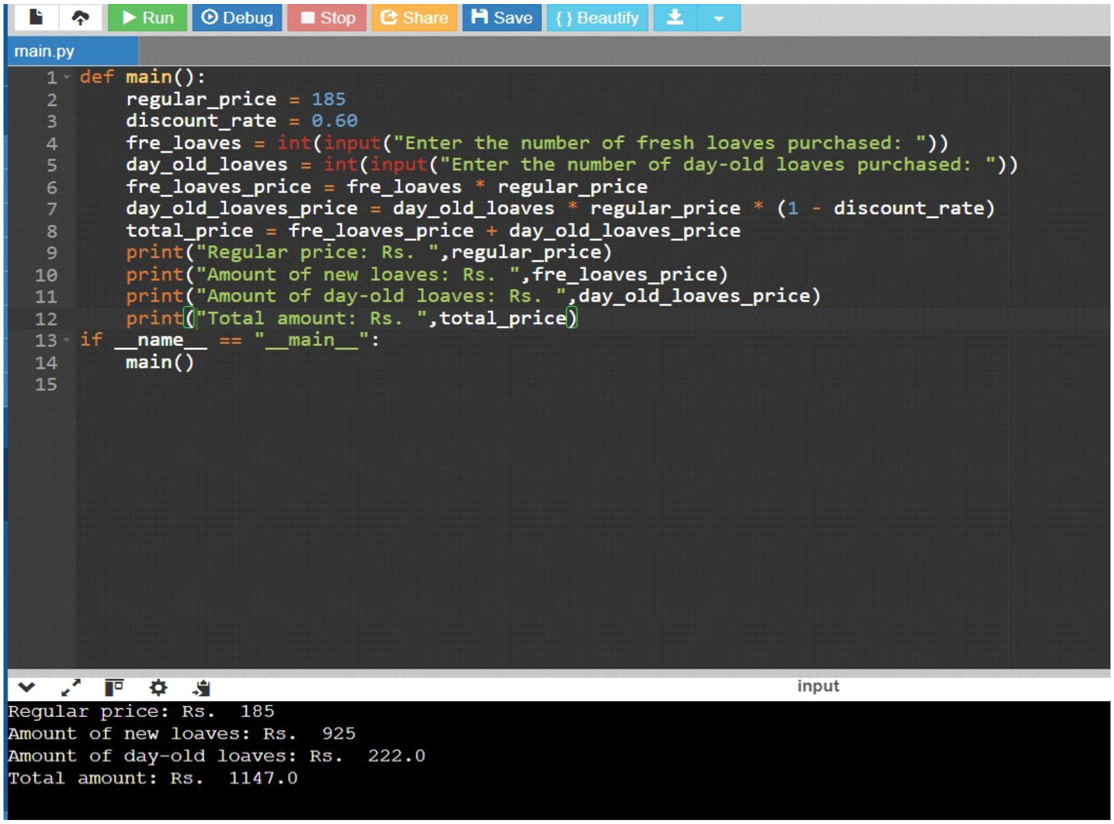
ASSISSMENT 3:

MALREDDY(192372015)

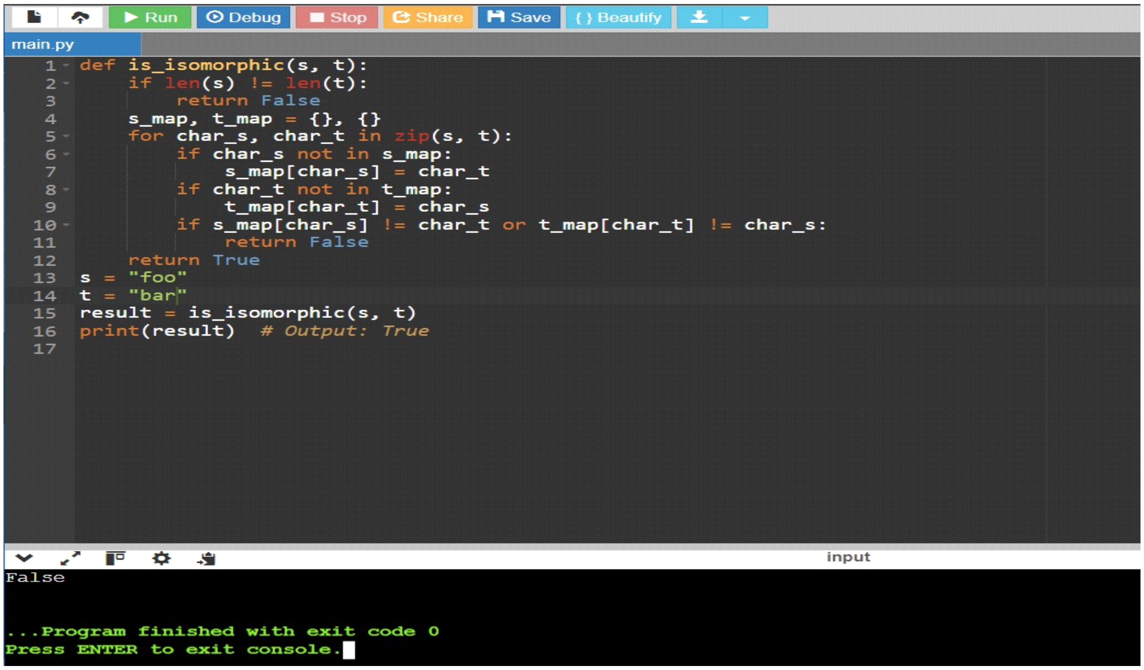
1.



```python
def main():
    regular_price = 185
    discount_rate = 0.60
    fre_loaves = int(input("Enter the number of fresh loaves purchased: "))
    day_old_loaves = int(input("Enter the number of day-old loaves purchased: "))
    fre_loaves_price = fre_loaves * regular_price
    day_old_loaves_price = day_old_loaves * regular_price * (1 - discount_rate)
    total_price = fre_loaves_price + day_old_loaves_price
    print("Regular price: Rs. ",regular_price)
    print("Amount of new loaves: Rs. ",fre_loaves_price)
    print("Amount of day-old loaves: Rs. ",day_old_loaves_price)
    print("Total amount: Rs. ",total_price)
if __name__ == "__main__":
    main()
```

```
Regular price: Rs.  185
Amount of new loaves: Rs.  925
Amount of day-old loaves: Rs.  222.0
Total amount: Rs.  1147.0
```

2.

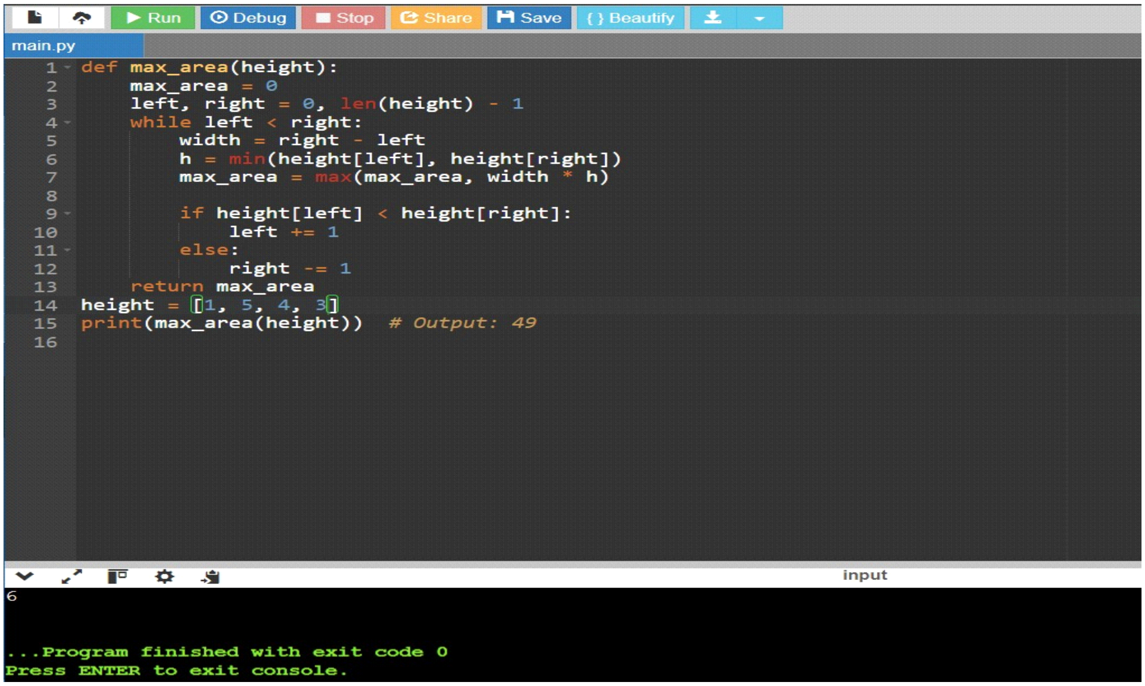


```python
def is_isomorphic(s, t):
    if len(s) != len(t):
        return False
    s_map, t_map = {}, {}
    for char_s, char_t in zip(s, t):
        if char_s not in s_map:
            s_map[char_s] = char_t
        if char_t not in t_map:
            t_map[char_t] = char_s
        if s_map[char_s] != char_t or t_map[char_t] != char_s:
            return False
    return True
s = "foo"
t = "bar"
result = is_isomorphic(s, t)
print(result)  # Output: True
```

```
False


...Program finished with exit code 0
Press ENTER to exit console.
```

3.



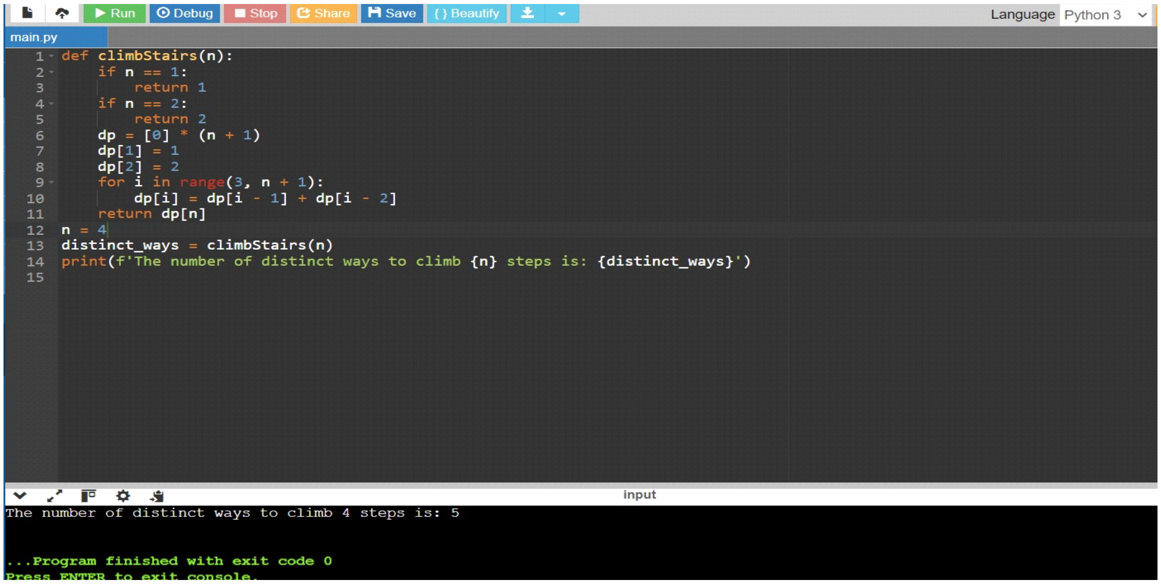
```python
def max_area(height):
    max_area = 0
    left, right = 0, len(height) - 1
    while left < right:
        width = right - left
        h = min(height[left], height[right])
        max_area = max(max_area, width * h)

        if height[left] < height[right]:
            left += 1
        else:
            right -= 1
    return max_area
height = [1, 5, 4, 3]
print(max_area(height))  # Output: 49
```

```
6


...Program finished with exit code 0
Press ENTER to exit console.
```

4.

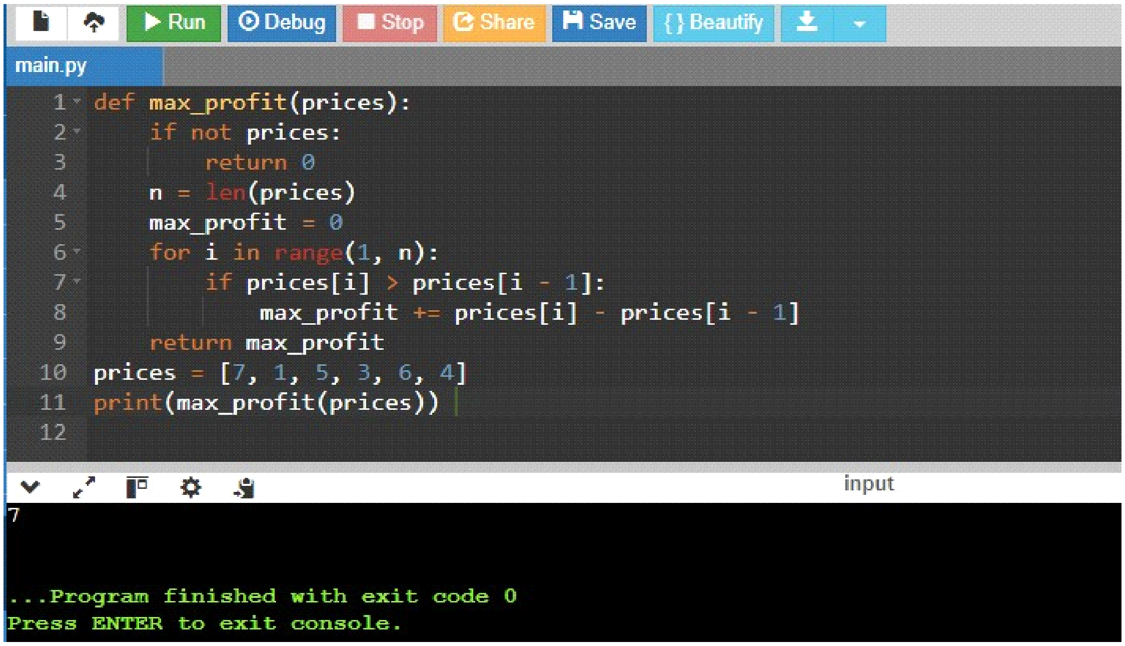


```python
def climbStairs(n):
    if n == 1:
        return 1
    if n == 2:
        return 2
    dp = [0] * (n + 1)
    dp[1] = 1
    dp[2] = 2
    for i in range(3, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]
    return dp[n]
n = 4
distinct_ways = climbStairs(n)
print(f'The number of distinct ways to climb {n} steps is: {distinct_ways}')
```

```
The number of distinct ways to climb 4 steps is: 5


...Program finished with exit code 0
Press ENTER to exit console.
```

5.



```python
def max_profit(prices):
    if not prices:
        return 0
    n = len(prices)
    max_profit = 0
    for i in range(1, n):
        if prices[i] > prices[i - 1]:
            max_profit += prices[i] - prices[i - 1]
    return max_profit
prices = [7, 1, 5, 3, 6, 4]
print(max_profit(prices))
```

```
7


...Program finished with exit code 0
Press ENTER to exit console.
```