

1. A bakery sells loaves of bread for 185 rupees each. Day old bread is discounted by 60 percent. Write a python program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. All of the values should be displayed using two decimal places, and the decimal points in all of the numbers should be aligned when reasonable values are entered by the user.

Sample Input:

Enter the number of fresh loves purchased: 5

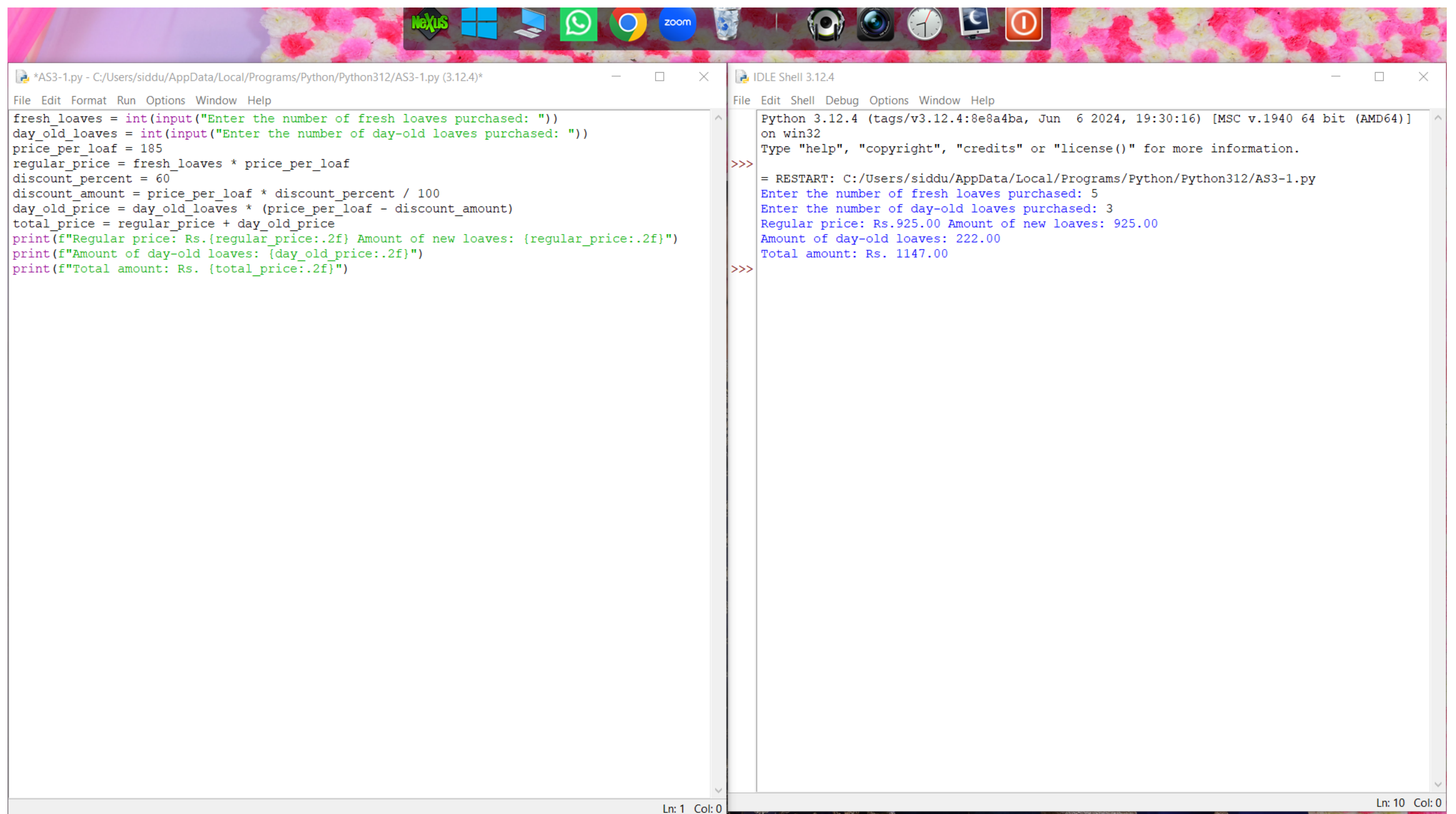
Enter the number of day-old loaves purchased: 3

Sample Output:

Regular price: Rs.185.00 Amount of new loaves: 925.00

Amount of day-old loaves: 333.00

Total amount: Rs. 1258.00



```
*AS3-1.py - C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-1.py (3.12.4)*
File Edit Format Run Options Window Help
fresh_loaves = int(input("Enter the number of fresh loaves purchased: "))
day_old_loaves = int(input("Enter the number of day-old loaves purchased: "))
price_per_loaf = 185
regular_price = fresh_loaves * price_per_loaf
discount_percent = 60
discount_amount = price_per_loaf * discount_percent / 100
day_old_price = day_old_loaves * (price_per_loaf - discount_amount)
total_price = regular_price + day_old_price
print(f"Regular price: Rs.{regular_price:.2f} Amount of new loaves: {regular_price:.2f}")
print(f"Amount of day-old loaves: {day_old_price:.2f}")
print(f"Total amount: Rs. {total_price:.2f}")

Python 3.12.4 (tags/v3.12.4:8e9a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-1.py
Enter the number of fresh loaves purchased: 5
Enter the number of day-old loaves purchased: 3
Regular price: Rs.925.00 Amount of new loaves: 925.00
Amount of day-old loaves: 222.00
Total amount: Rs. 1147.00
>>>
```

2. Given two strings “ s” and “ t” , determine if they are isom orphic. Two strings “ s” and “ t” are isomorphic if the characters in “ s” can be replaced to get “ t” . All occurrences of a character must be replaced with another character while preserving the order of characters.

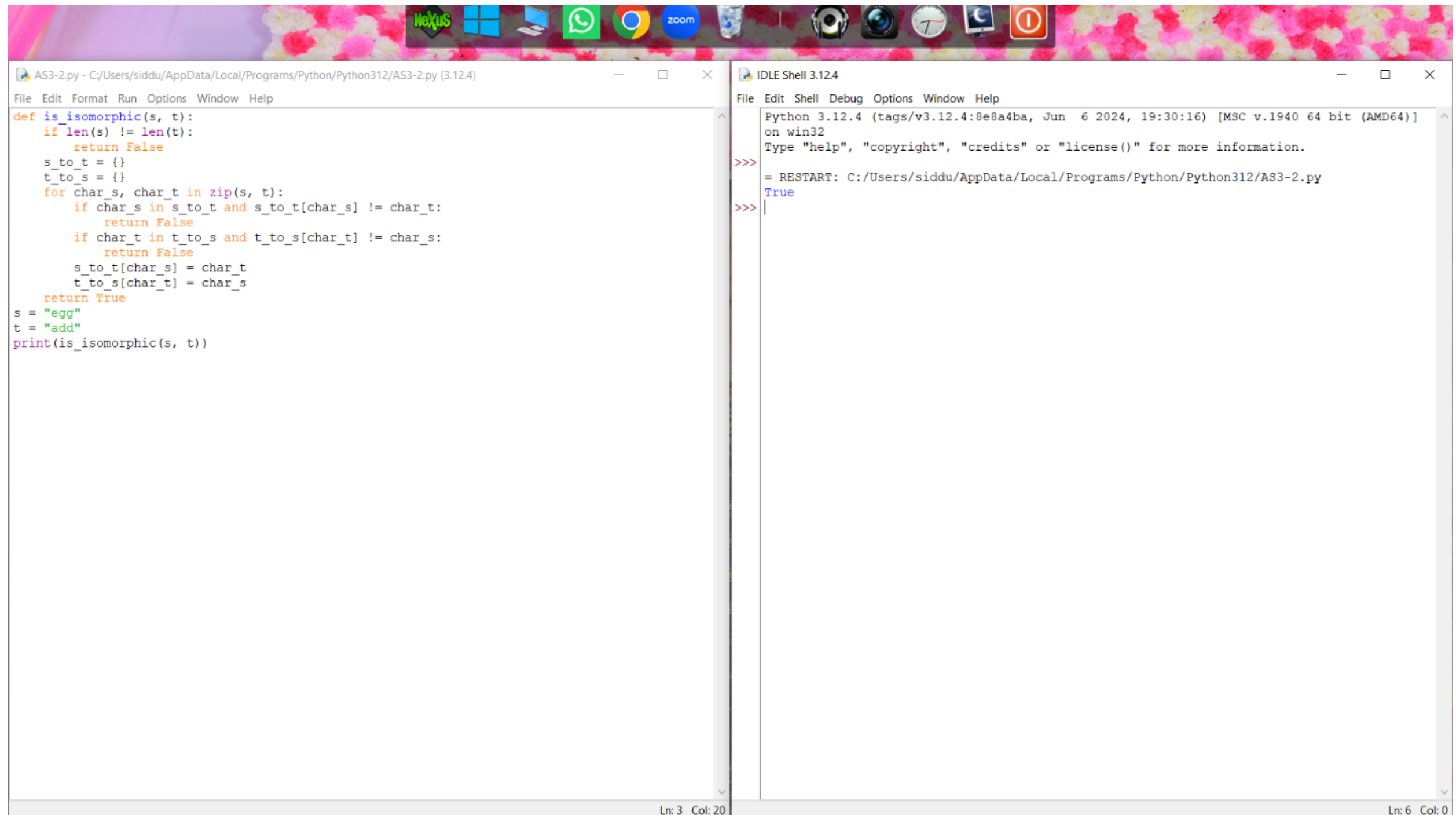
No two characters may map to the same character, but a character may map to itself.

Constraints:

✓ s and t consist of any valid ascii character.

Test Cases:

Input: s = "egg", t = "add" Output: true



The screenshot shows a Python IDE with two windows. The left window, titled 'AS3-2.py', contains the following code:

```
def is_isomorphic(s, t):  
    if len(s) != len(t):  
        return False  
    s_to_t = {}  
    t_to_s = {}  
    for char_s, char_t in zip(s, t):  
        if char_s in s_to_t and s_to_t[char_s] != char_t:  
            return False  
        if char_t in t_to_s and t_to_s[char_t] != char_s:  
            return False  
        s_to_t[char_s] = char_t  
        t_to_s[char_t] = char_s  
    return True  
s = "egg"  
t = "add"  
print(is_isomorphic(s, t))
```

The right window, titled 'IDLE Shell 3.12.4', shows the output of the script:

```
Python 3.12.4 (tags/v3.12.4:9e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-2.py  
True  
>>> |
```

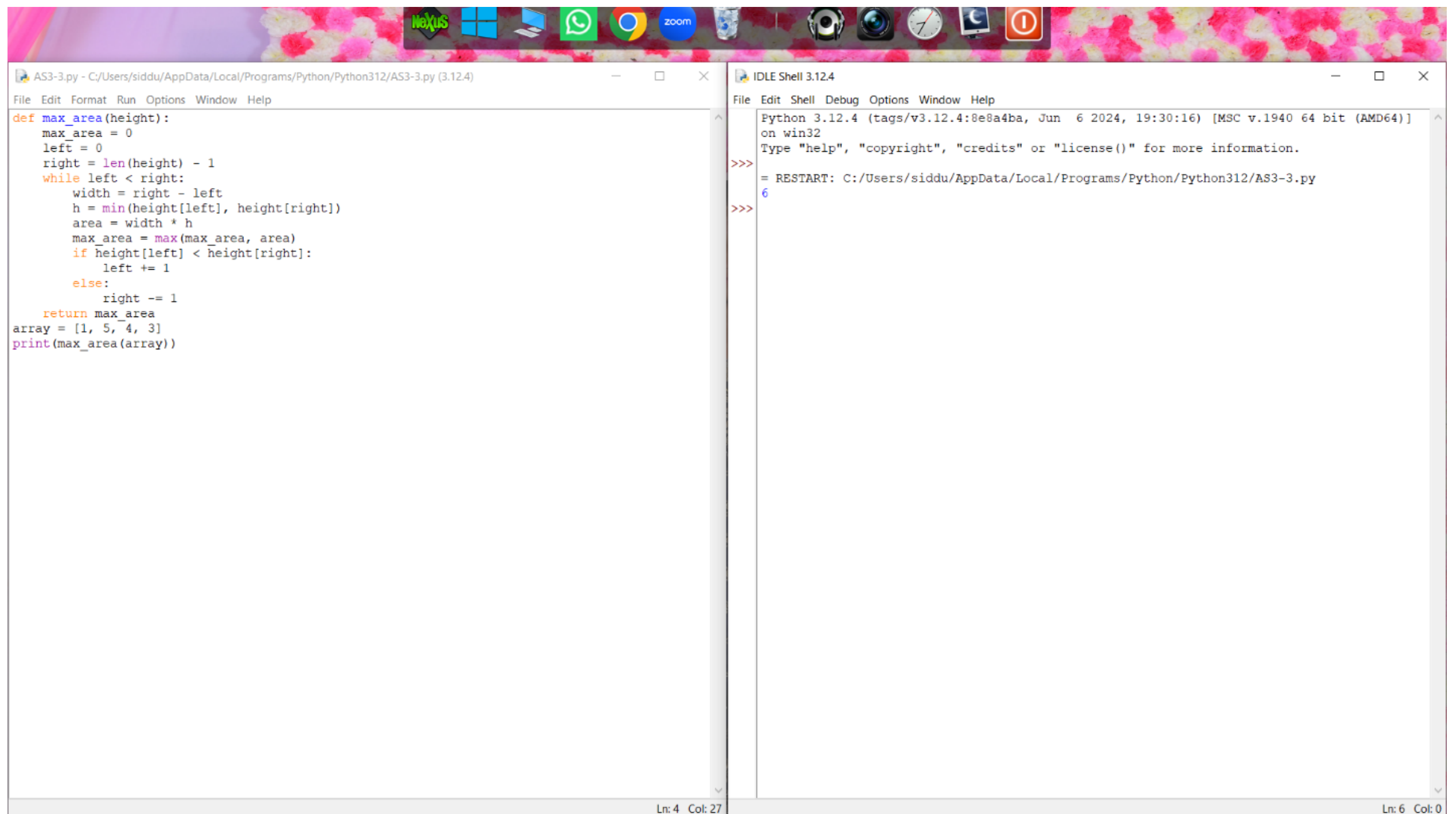
3. Given  $n$  non-negative integers  $a_1, a_2, a_3, \dots, a_n$  where each represents a point at coordinate  $(i, a_i)$ . ' $n$ ' vertical lines are drawn such that the two endpoints of line  $i$  is at  $(i, a_i)$  and  $(i, 0)$ . Find two lines, which together with x-axis forms a container, such that the container contains the most water. The program should return an integer which corresponds to the maximum area of water that can be contained (maximum area instead of maximum volume sounds weird but this is the 2D plane we are working with for simplicity).

Note:

You may not slant the container.

Test case:

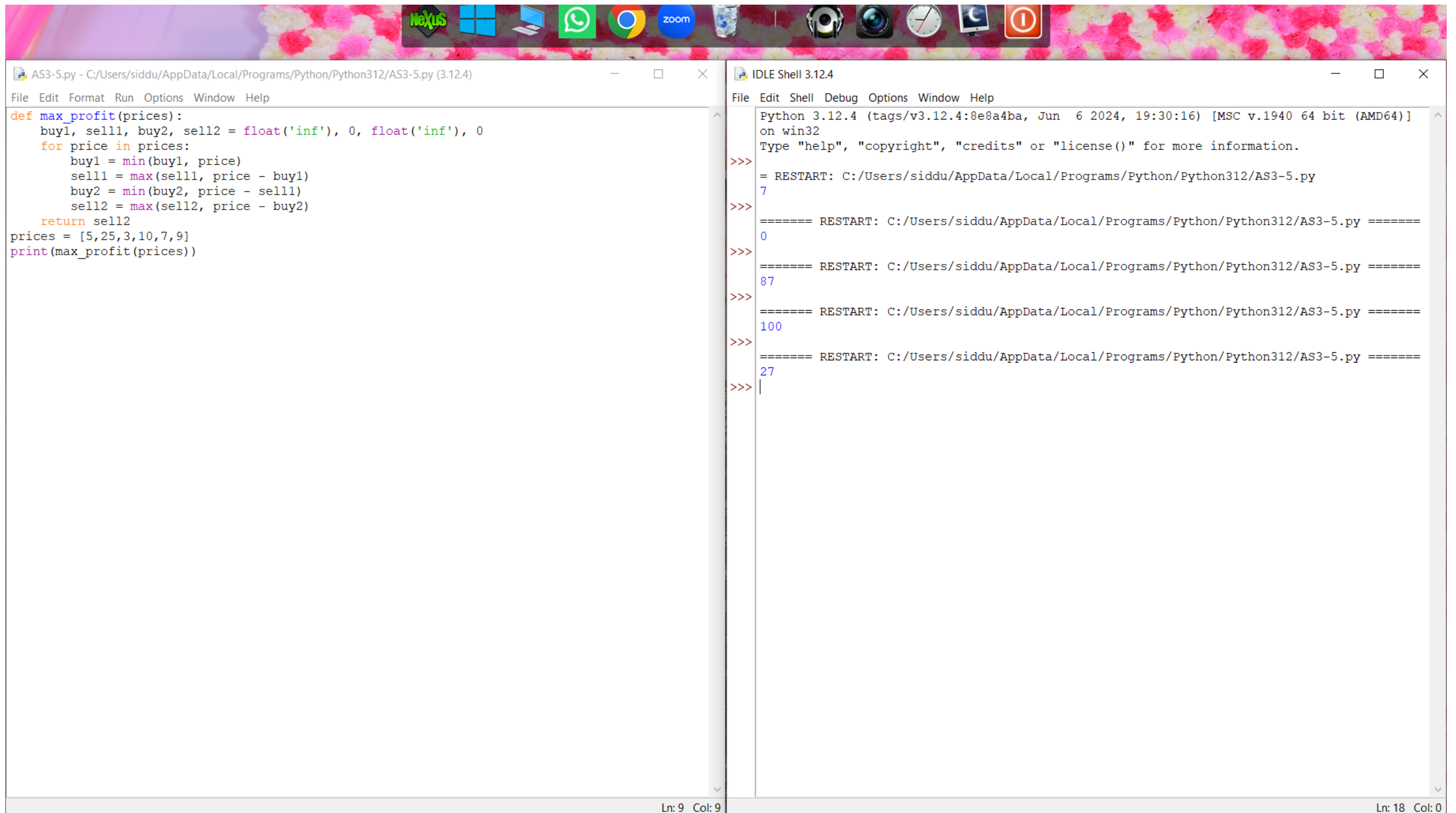
1. Input: array = [1, 5, 4, 3] Output: 6



4. You are climbing a staircase. It takes  $n$  steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Test Case:

1. Input:  $n = 2$  Output: 2
2. Input:  $n = 3$  Output: 3
3. Input:  $n = 4$  Output: 5
4. Input:  $n = 1$  Output: 1
5. Input:  $n = 5$  Output: 8



```
def max_profit(prices):
    buy1, sell1, buy2, sell2 = float('inf'), 0, float('inf'), 0
    for price in prices:
        buy1 = min(buy1, price)
        sell1 = max(sell1, price - buy1)
        buy2 = min(buy2, price - sell1)
        sell2 = max(sell2, price - buy2)
    return sell2
prices = [5,25,3,10,7,9]
print(max_profit(prices))
```

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py
7
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
0
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
87
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
100
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
27
>>> |
```

5. In daily share trading, a buyer buys shares in the morning and sells them on the same day.

If the trader is allowed to make at most 2 transactions in a day, whereas the second transaction can only start after the first one is complete (Buy->sell->Buy->sell). Given stock prices throughout the day, find out the maximum profit that a share trader could have made.

Test Case:

- 1.Input: prices = [7,1,5,3,6,4] Output: 7
- 2.Input: prices = [7,6,4,3,1] Output: 0
- 3.Input: [10, 22, 5, 75, 65, 80] Output:87
- 4.Input: [2, 30, 15, 10, 8, 25, 80] Output:100
- 5. Input: [5,25,3,10,7,9] Output:27

AS3-5.py - C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py (3.12.4)

File Edit Format Run Options Window Help

```
def max_profit(prices):
    buy1, sell1, buy2, sell2 = float('inf'), 0, float('inf'), 0
    for price in prices:
        buy1 = min(buy1, price)
        sell1 = max(sell1, price - buy1)
        buy2 = min(buy2, price - sell1)
        sell2 = max(sell2, price - buy2)
    return sell2
prices = [5,25,3,10,7,9]
print(max_profit(prices))
```

Ln: 9 Col: 9

IDLE Shell 3.12.4

File Edit Shell Debug Options Window Help

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py
7
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
0
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
87
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
100
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS3-5.py =====
27
>>> |
```

Ln: 18 Col: 0