SAVEETHA SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTERSCIENCE AND ENGINEERING

CSA0889 – Python Programming

Assignment – 3&4

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
   Test cases: 1. 4, 6 2. -1,5 3. 0,6 4. 7,8 5. 3,4

2. Given two strings “s” and “t”, determine if they are isomorphic. 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:

   1. Input: s = "egg", t = "add"   Output: true
   2. Input: s = "foo", t = "bar"   Output: false
   3. Input: s = "paper", t = "title"   Output: true
   4. Input: s = "fry", t = "sky"   Output: true
   5. Input: s = "apples", t = "apple"   Output: false

3. Given n non-negative integers a1, a2,a3,…an where each represents a point at coordinate (i, ai) . ‘ n ‘ vertical lines are drawn such that the two endpoints of line i is at (i, ai) 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
2. Input: array = [3, 1, 2, 4, 5] Output: 12
3. Input: array = [1,8,6,2,5,4,8,3,7] Output: 49
4. Input: array = [1,1] Output: 1
5. Input: array = [7,3] Output: 3

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

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:

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

6. Given an integer n, return the number of strings of length n that consist only of vowels (a, e, i, o, u) and are lexicographically sorted. A string s is lexicographically sorted if for all valid i, s[i] is the same as or comes before s[i+1] in the alphabet.

Test Cases:

1. Input: n = 1 Output: 5

Explanation: The 5 sorted strings that consist of vowels only are ["a","e","i","o","u"].

2. Input: n = 2 Output: 15 Explanation: The 15 sorted strings that consist of vowels only are ["aa","ae","ai","ao","au","ee","ei","eo","eu","ii","io","iu","oo","ou","uu"]. Note that "ea" is not a valid string since 'e' comes after 'a' in the alphabet.
3. Input: n = 33 Output: 66045
4. n=-5 5.n=10

7. Given two binary strings a and b, return their sum as a binary string.

- a and b consist only of '0' or '1' characters.
- Each string does not contain leading zeros except for the zero itself.

Test cases:

1. Input: a = "11", b = "1" Output: "100"
2. Input: a = "1010", b = "1011" Output: "10101"
3. a= “1111”, b= “1010”
4. a= “101101”, b= “1100” 5.a= “1011” b= “1111”

8. Basic Calculator II Given a string s which represents an expression, evaluate this expression and return its value. The integer division should truncate toward zero. You may assume that the given expression is always valid. All intermediate results will be in the range of [-231, 231 - 1].

- s consists of integers and operators ('+', '-', '*', '/') separated by some number of spaces.
- s represents a valid expression.
- All the integers in the expression are non-negative integers in the range [0, 231 - 1].

The answer is guaranteed to fit in a 32-bit integer.

Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval().

Test cases:

1. Input: s = "3+2*2" Output: 7
2. Input: s = " 3/2 " Output: 1
3. Input: s = " 3+5 / 2 " Output: 5
4. s= “-1+5” 5.s= “2+3+5”

9. Raju, has again started troubling people in your city. The people have turned on to you for getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9 characters. He wants you to reverse it from the final answer such that the number becomes Mirror number. A Mirror is a number which equals its reverse. The hope of people are on you so you have to solve the riddle. You have to tell if some number exists which you would reverse to convert the number into Mirror

Sample input:

Enter the number: 123456

Sample output: Mirror image: 654321

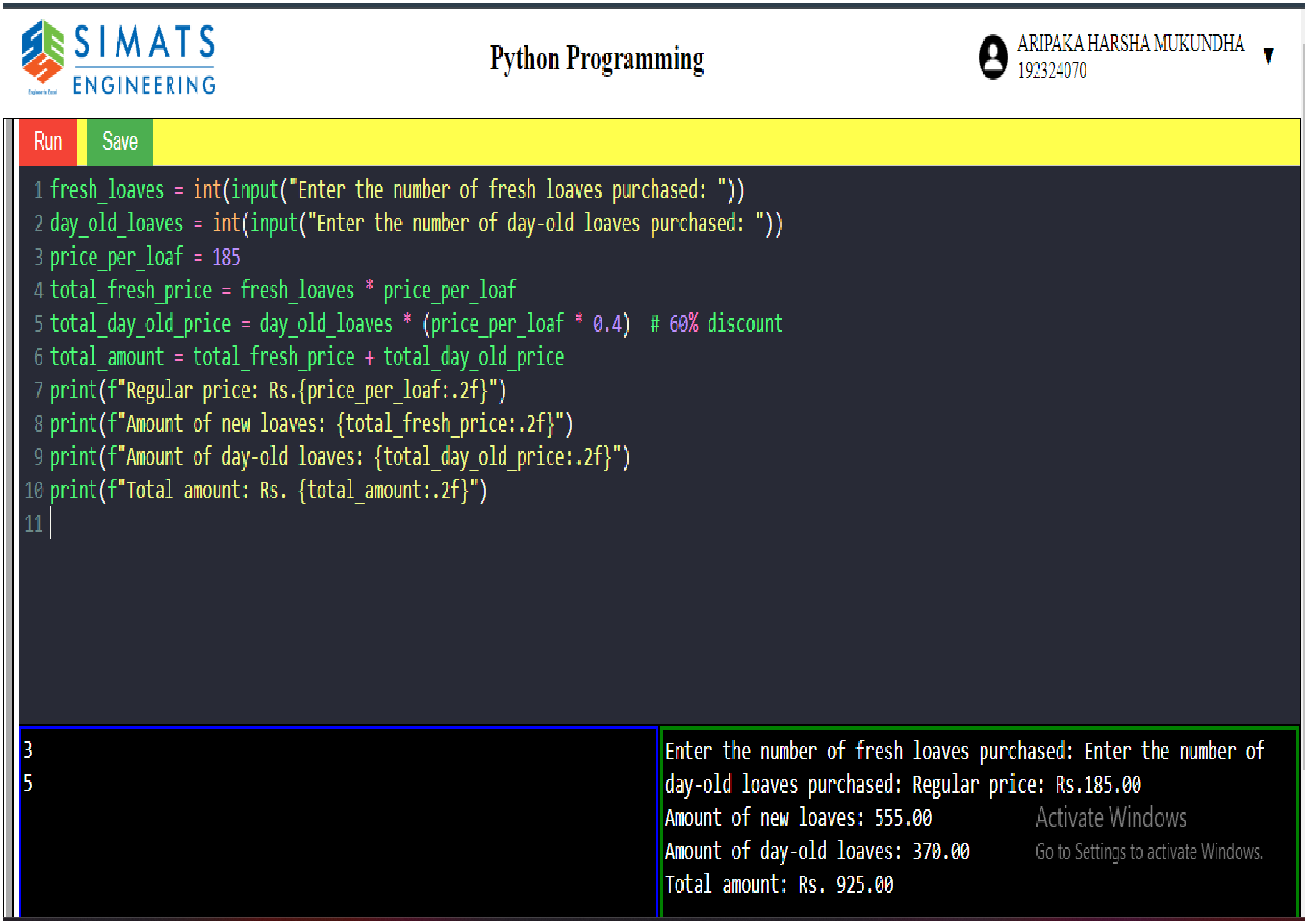
Test cases:

1. Sell123
2. 5489236
3. Abc-abc
4. %$$$$^&
5. -123456

10. Write a python function called matches that takes two strings as arguments and returns how many matches there are between the strings. A match is where the two strings have the same character at the same index.

    Test Cases:
    1. Input: s1= “what” s2= “watch” Output: 1
    2. Input: s1= “ ran” s2= “van”
    3. Input : s1 = “ rain” s2 = “ turn”
    4. Input : s1 = “ python” s2 = “py”
    5. Inpput: s1= “man” s2= “women

1)



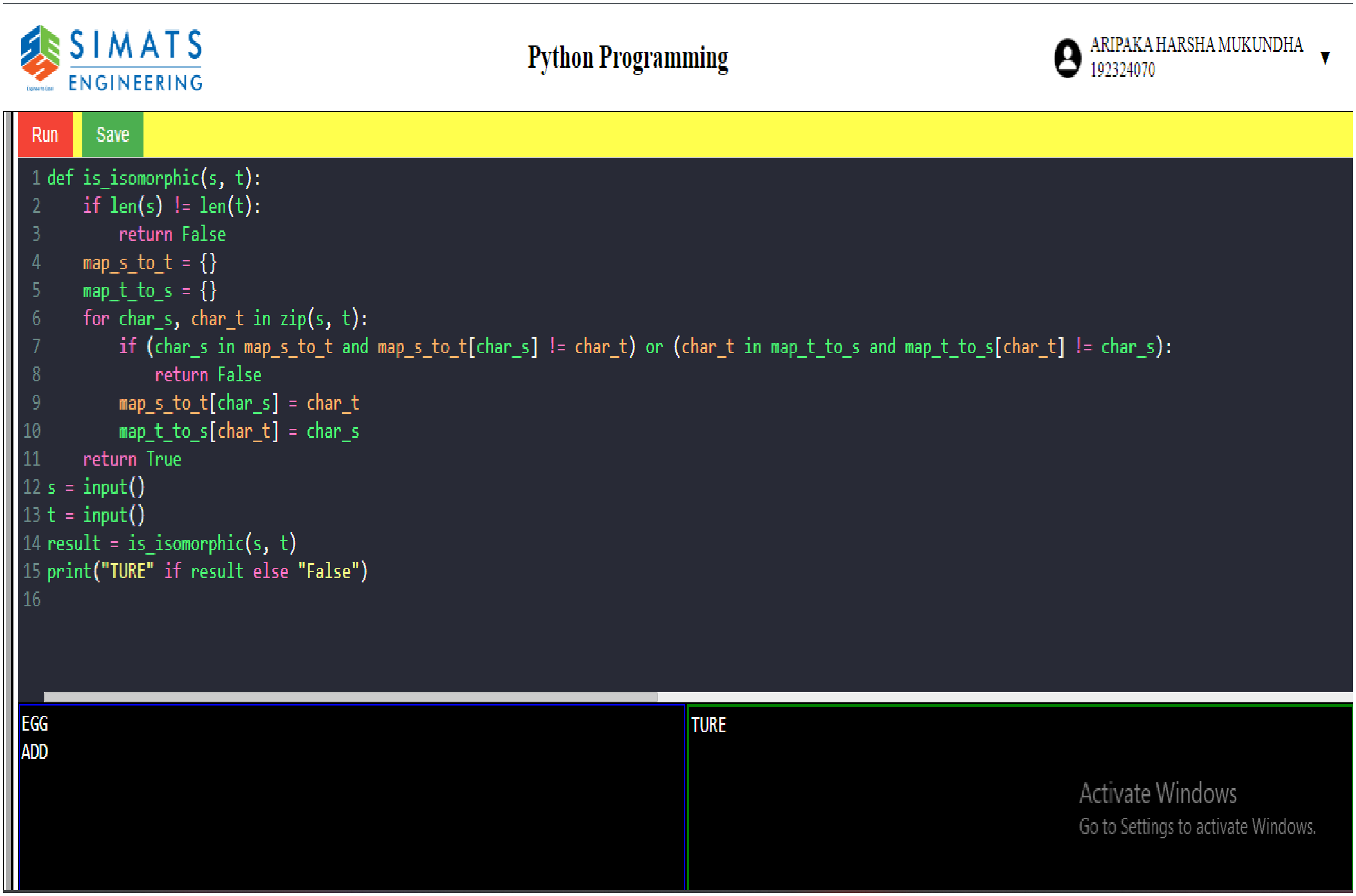
```python
fresh_loaves = int(input("Enter the number of fresh loaves purchased: "))
day_old_loaves = int(input("Enter the number of day-old loaves purchased: "))
price_per_loaf = 185
total_fresh_price = fresh_loaves * price_per_loaf
total_day_old_price = day_old_loaves * (price_per_loaf * 0.4)  # 60% discount
total_amount = total_fresh_price + total_day_old_price
print(f"Regular price: Rs.{price_per_loaf:.2f}")
print(f"Amount of new loaves: {total_fresh_price:.2f}")
print(f"Amount of day-old loaves: {total_day_old_price:.2f}")
print(f"Total amount: Rs. {total_amount:.2f}")

```

```
3
5
```

```
Enter the number of fresh loaves purchased: Enter the number of
day-old loaves purchased: Regular price: Rs.185.00
Amount of new loaves: 555.00
Amount of day-old loaves: 370.00
Total amount: Rs. 925.00
```

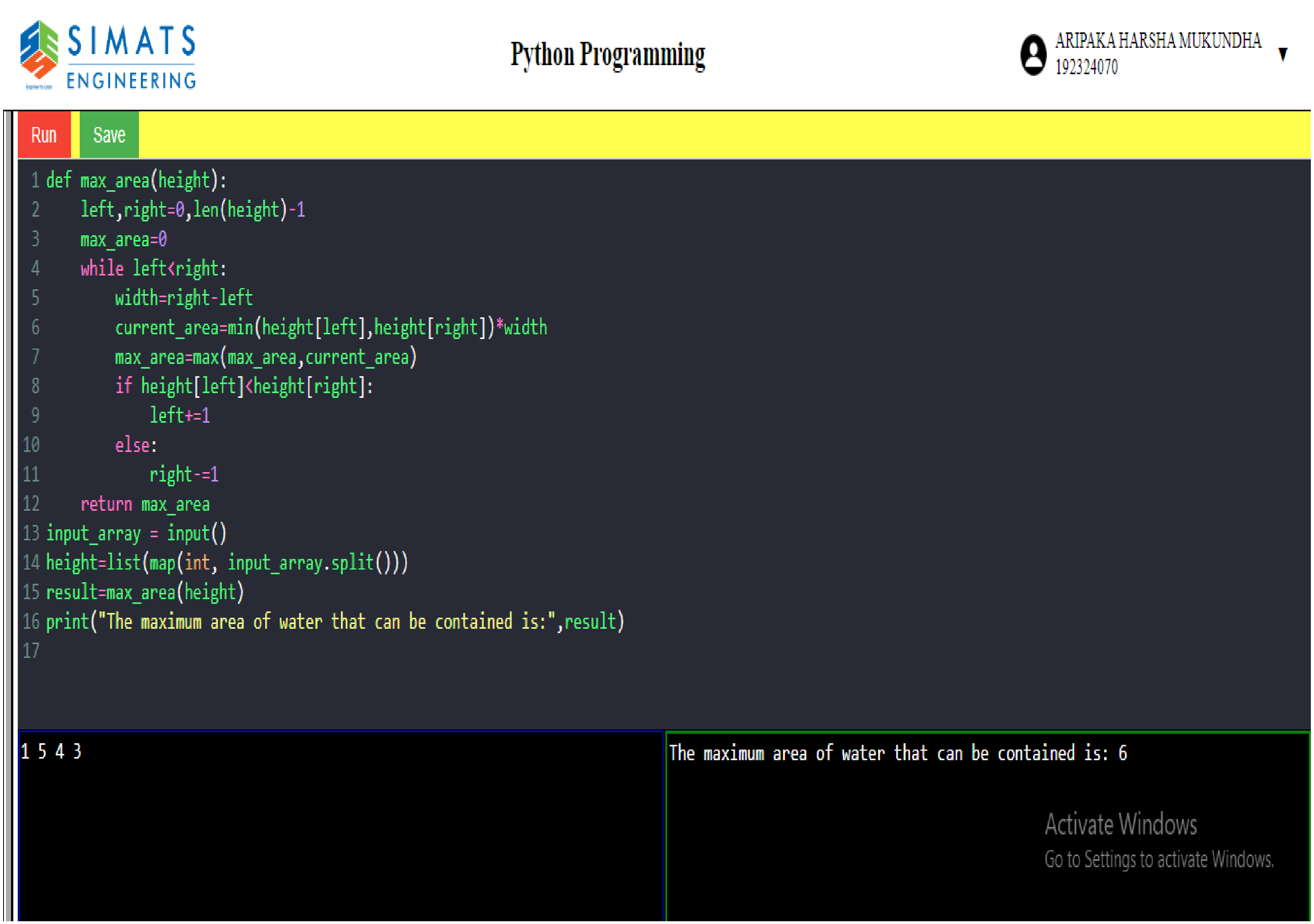
2)



```python
def is_isomorphic(s, t):
    if len(s) != len(t):
        return False
    map_s_to_t = {}
    map_t_to_s = {}
    for char_s, char_t in zip(s, t):
        if (char_s in map_s_to_t and map_s_to_t[char_s] != char_t) or (char_t in map_t_to_s and map_t_to_s[char_t] != char_s):
            return False
        map_s_to_t[char_s] = char_t
        map_t_to_s[char_t] = char_s
    return True
s = input()
t = input()
result = is_isomorphic(s, t)
print("TURE" if result else "False")
```

```
EGG
ADD
```

```
TURE
```



Python Programming

ARIPAKA HARSHA MUKUNDHA
192324070

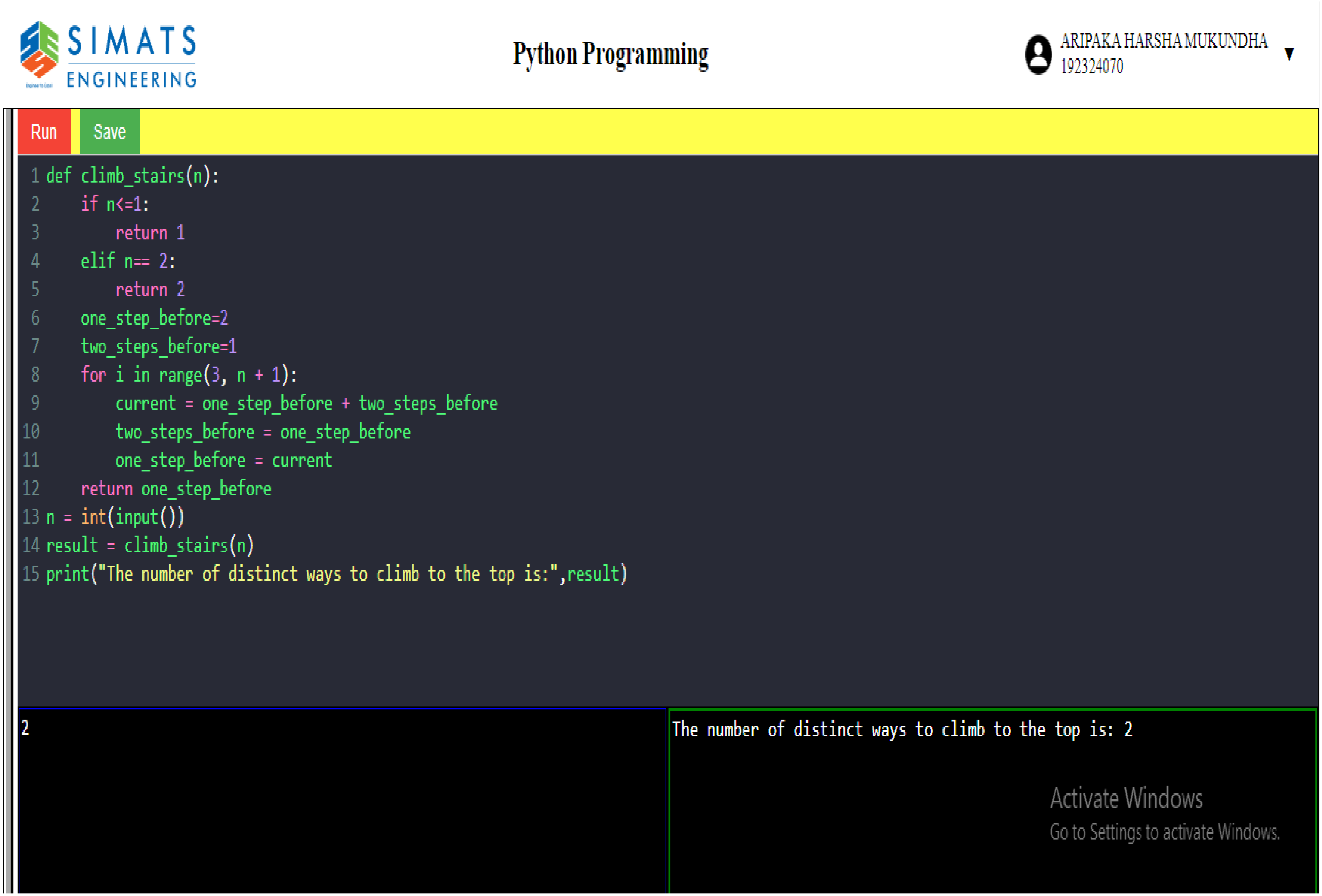
```python
def max_area(height):
    left,right=0,len(height)-1
    max_area=0
    while left<right:
        width=right-left
        current_area=min(height[left],height[right])*width
        max_area=max(max_area,current_area)
        if height[left]<height[right]:
            left+=1
        else:
            right-=1
    return max_area
input_array = input()
height=list(map(int, input_array.split()))
result=max_area(height)
print("The maximum area of water that can be contained is:",result)
```

```
1 5 4 3
```

```
The maximum area of water that can be contained is: 6
```

3)

4)

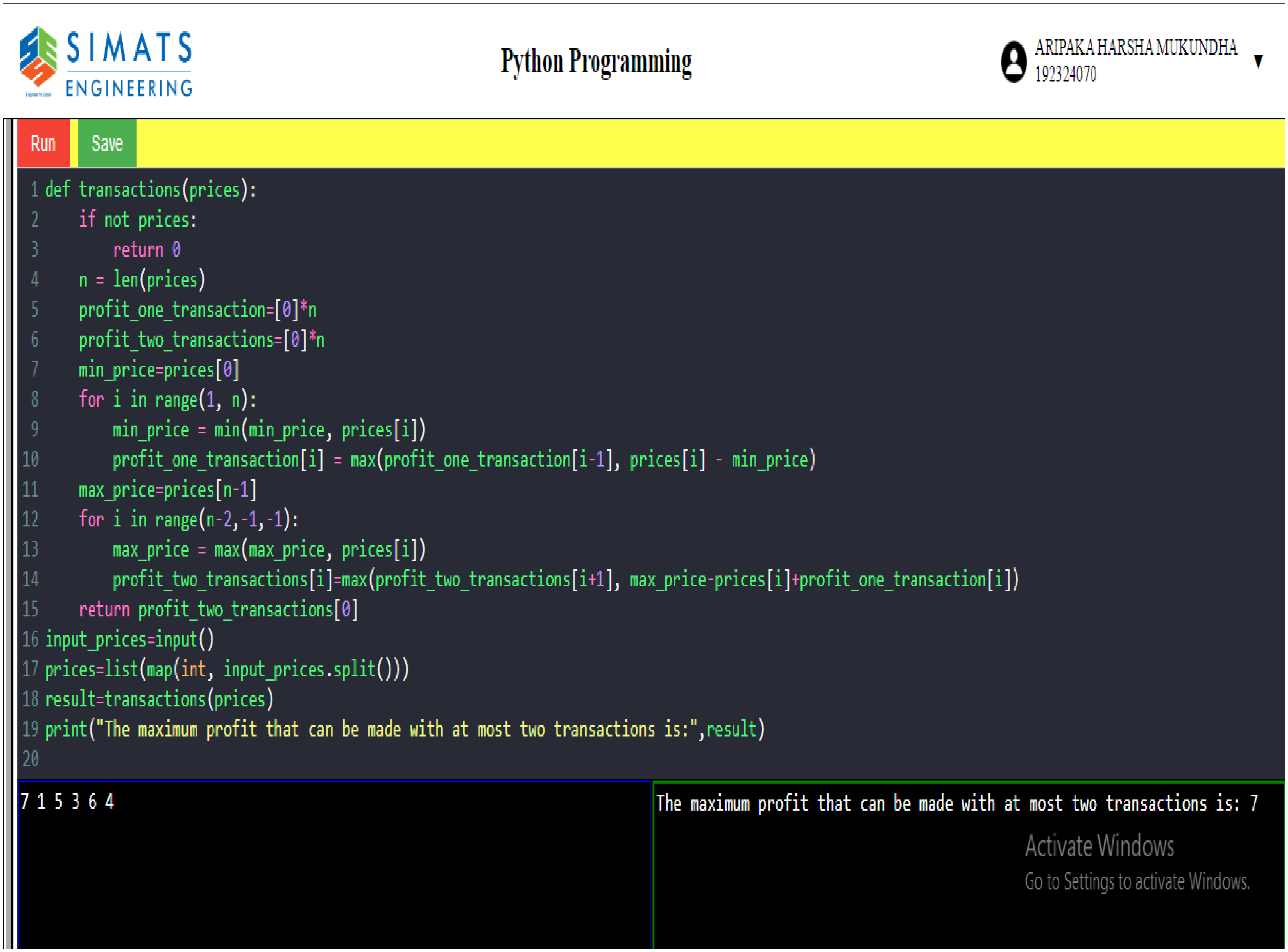


```
def climb_stairs(n):
    if n<=1:
        return 1
    elif n== 2:
        return 2
    one_step_before=2
    two_steps_before=1
    for i in range(3, n + 1):
        current = one_step_before + two_steps_before
        two_steps_before = one_step_before
        one_step_before = current
    return one_step_before
n = int(input())
result = climb_stairs(n)
print("The number of distinct ways to climb to the top is:",result)
```

```
2
```

```
The number of distinct ways to climb to the top is: 2
```

5)

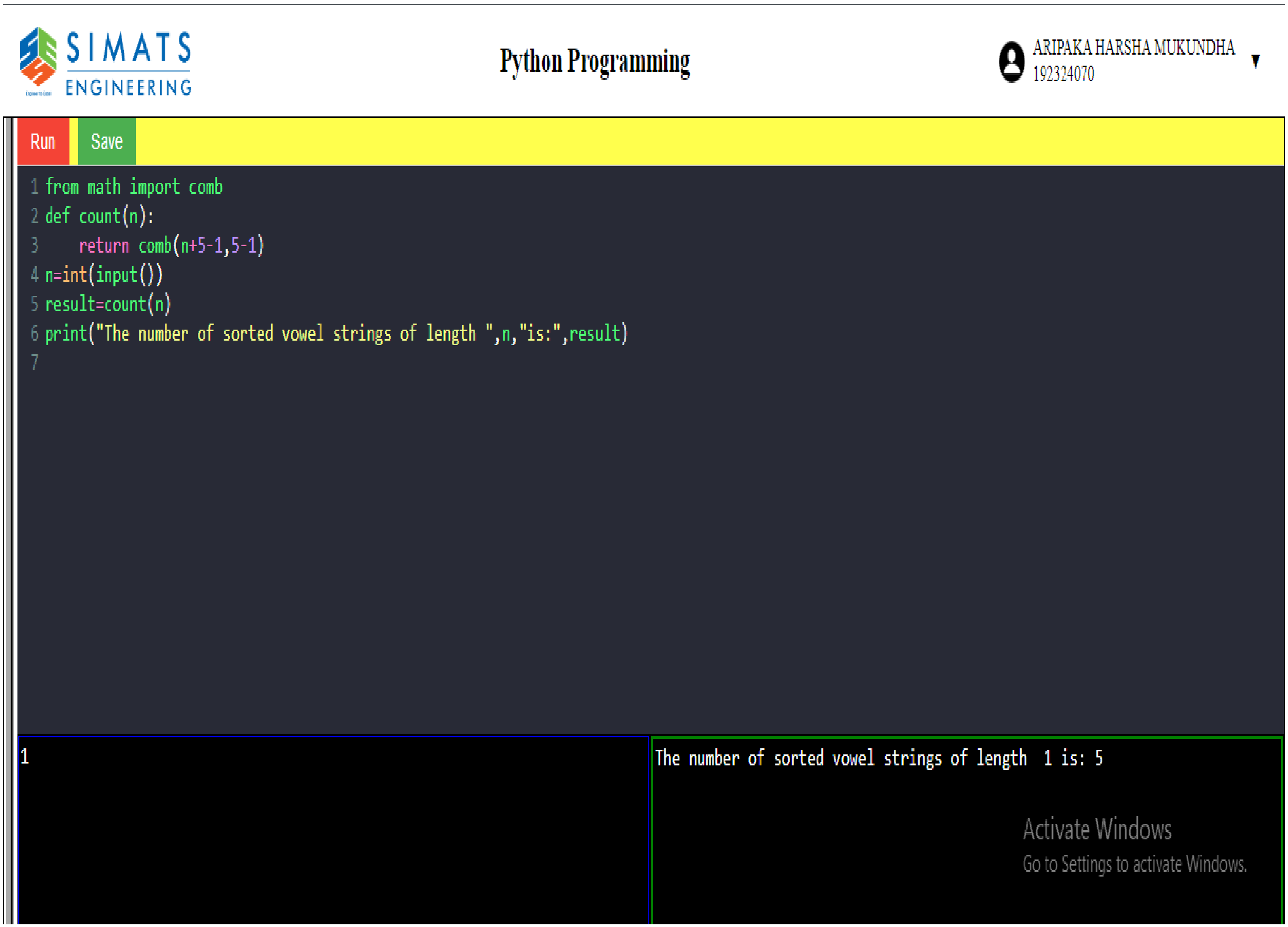


```
def transactions(prices):
    if not prices:
        return 0
    n = len(prices)
    profit_one_transaction=[0]*n
    profit_two_transactions=[0]*n
    min_price=prices[0]
    for i in range(1, n):
        min_price = min(min_price, prices[i])
        profit_one_transaction[i] = max(profit_one_transaction[i-1], prices[i] - min_price)
    max_price=prices[n-1]
    for i in range(n-2,-1,-1):
        max_price = max(max_price, prices[i])
        profit_two_transactions[i]=max(profit_two_transactions[i+1], max_price-prices[i]+profit_one_transaction[i])
    return profit_two_transactions[0]
input_prices=input()
prices=list(map(int, input_prices.split()))
result=transactions(prices)
print("The maximum profit that can be made with at most two transactions is:",result)
```

```
7 1 5 3 6 4
```

```
The maximum profit that can be made with at most two transactions is: 7
```

6)

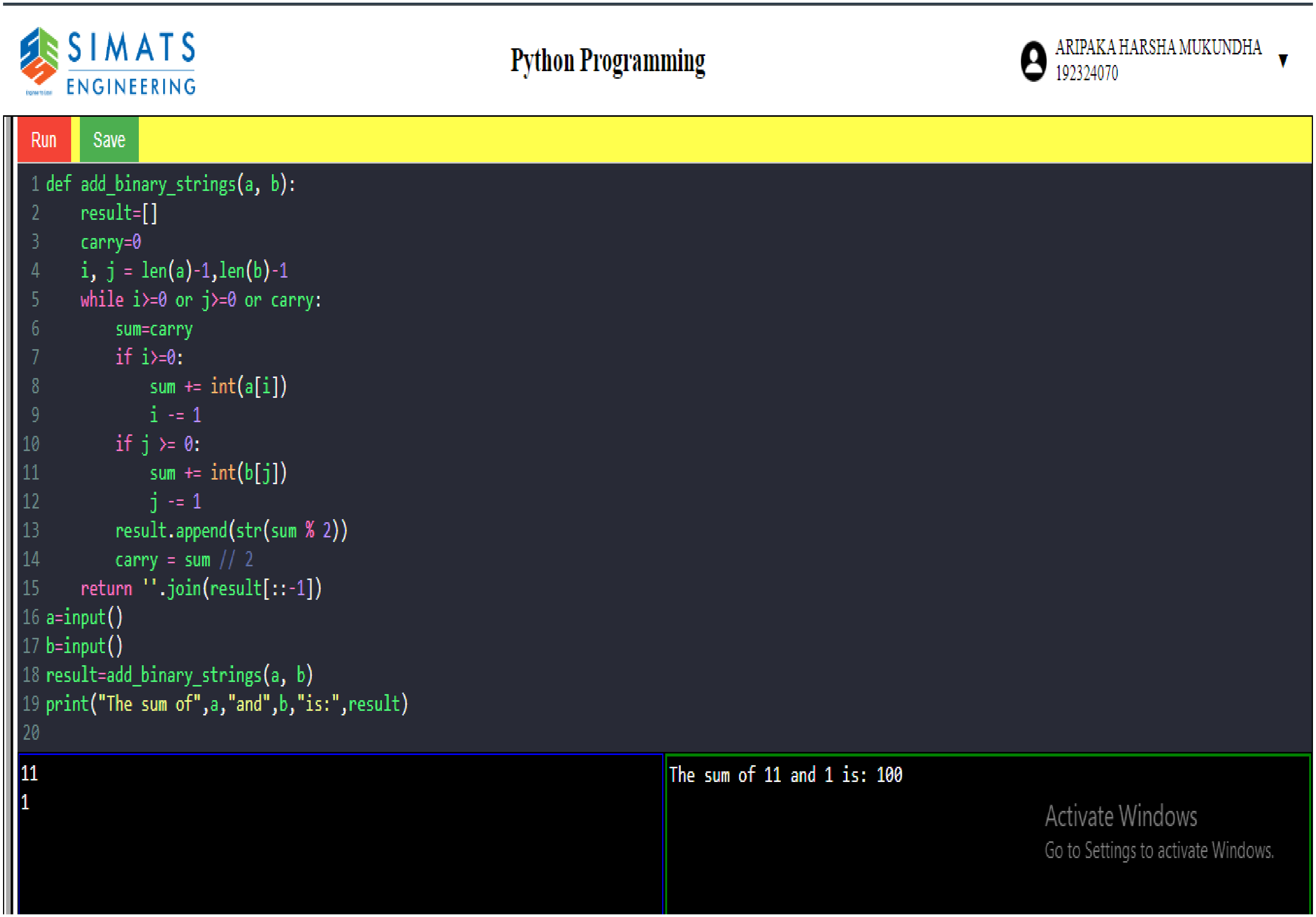


```python
from math import comb
def count(n):
    return comb(n+5-1,5-1)
n=int(input())
result=count(n)
print("The number of sorted vowel strings of length ",n,"is:",result)
```

```
1
```

```
The number of sorted vowel strings of length  1 is: 5
```

7)

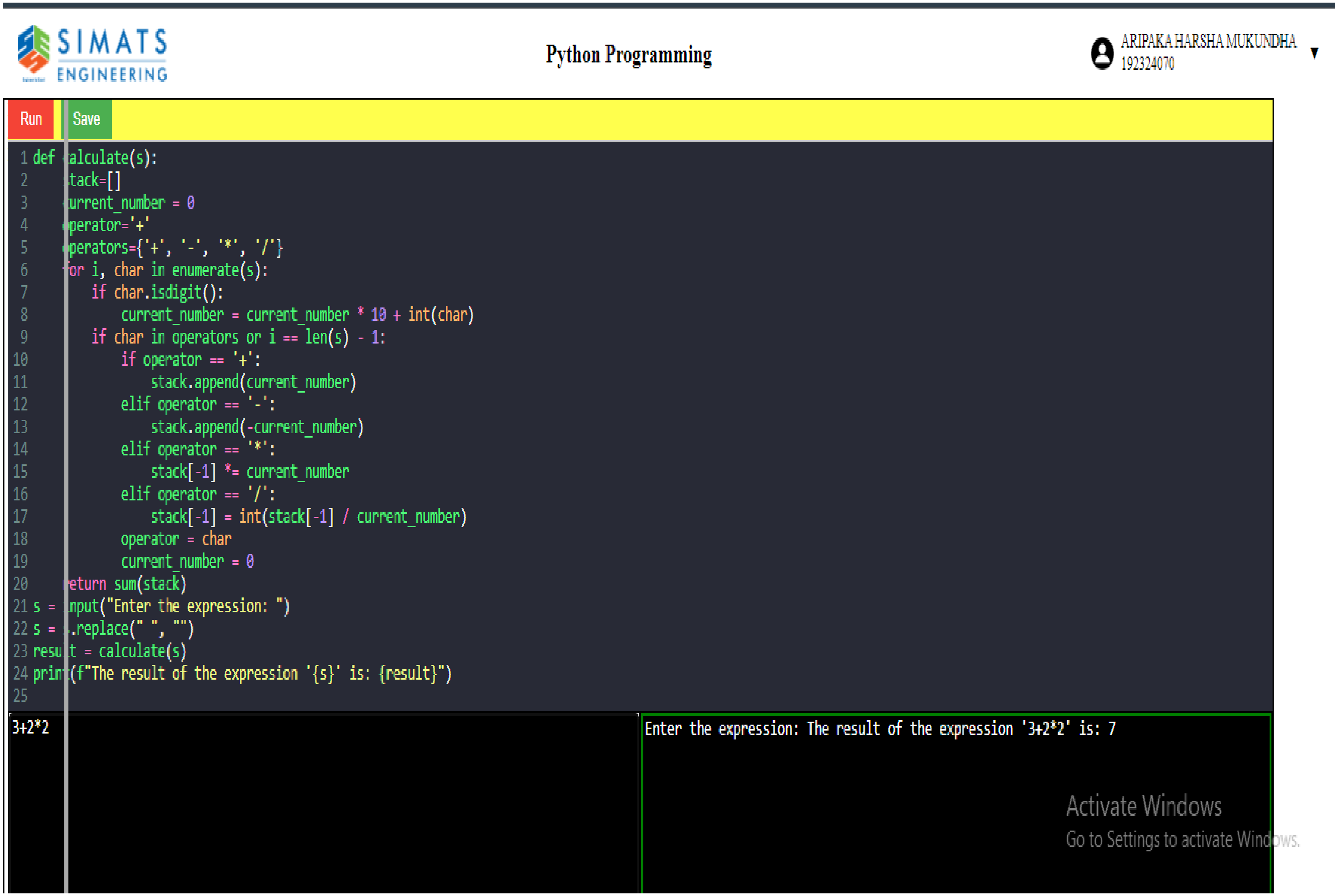


```python
def add_binary_strings(a, b):
    result=[]
    carry=0
    i, j = len(a)-1,len(b)-1
    while i>=0 or j>=0 or carry:
        sum=carry
        if i>=0:
            sum += int(a[i])
            i -= 1
        if j >= 0:
            sum += int(b[j])
            j -= 1
        result.append(str(sum % 2))
        carry = sum // 2
    return ''.join(result[::-1])
a=input()
b=input()
result=add_binary_strings(a, b)
print("The sum of",a,"and",b,"is:",result)
```

```
11
1
```

```
The sum of 11 and 1 is: 100
```

8)

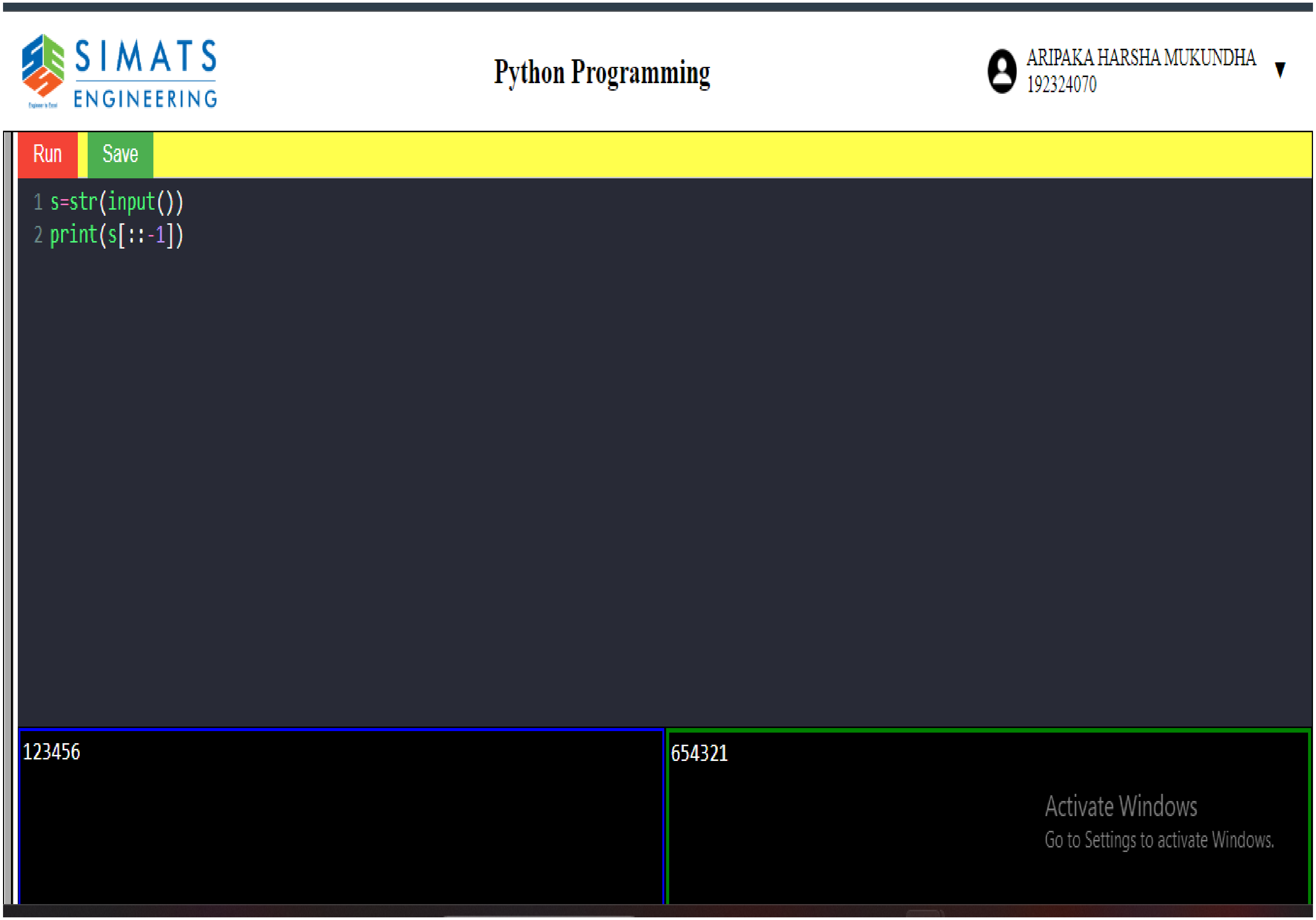


```python
def calculate(s):
    stack=[]
    current_number = 0
    operator='+'
    operators={'+', '-', '*', '/'}
    for i, char in enumerate(s):
        if char.isdigit():
            current_number = current_number * 10 + int(char)
        if char in operators or i == len(s) - 1:
            if operator == '+':
                stack.append(current_number)
            elif operator == '-':
                stack.append(-current_number)
            elif operator == '*':
                stack[-1] *= current_number
            elif operator == '/':
                stack[-1] = int(stack[-1] / current_number)
            operator = char
            current_number = 0
    return sum(stack)
s = input("Enter the expression: ")
s = s.replace(" ", "")
result = calculate(s)
print(f"The result of the expression '{s}' is: {result}")
```

```
3+2*2
```

```
Enter the expression: The result of the expression '3+2*2' is: 7
```

9)

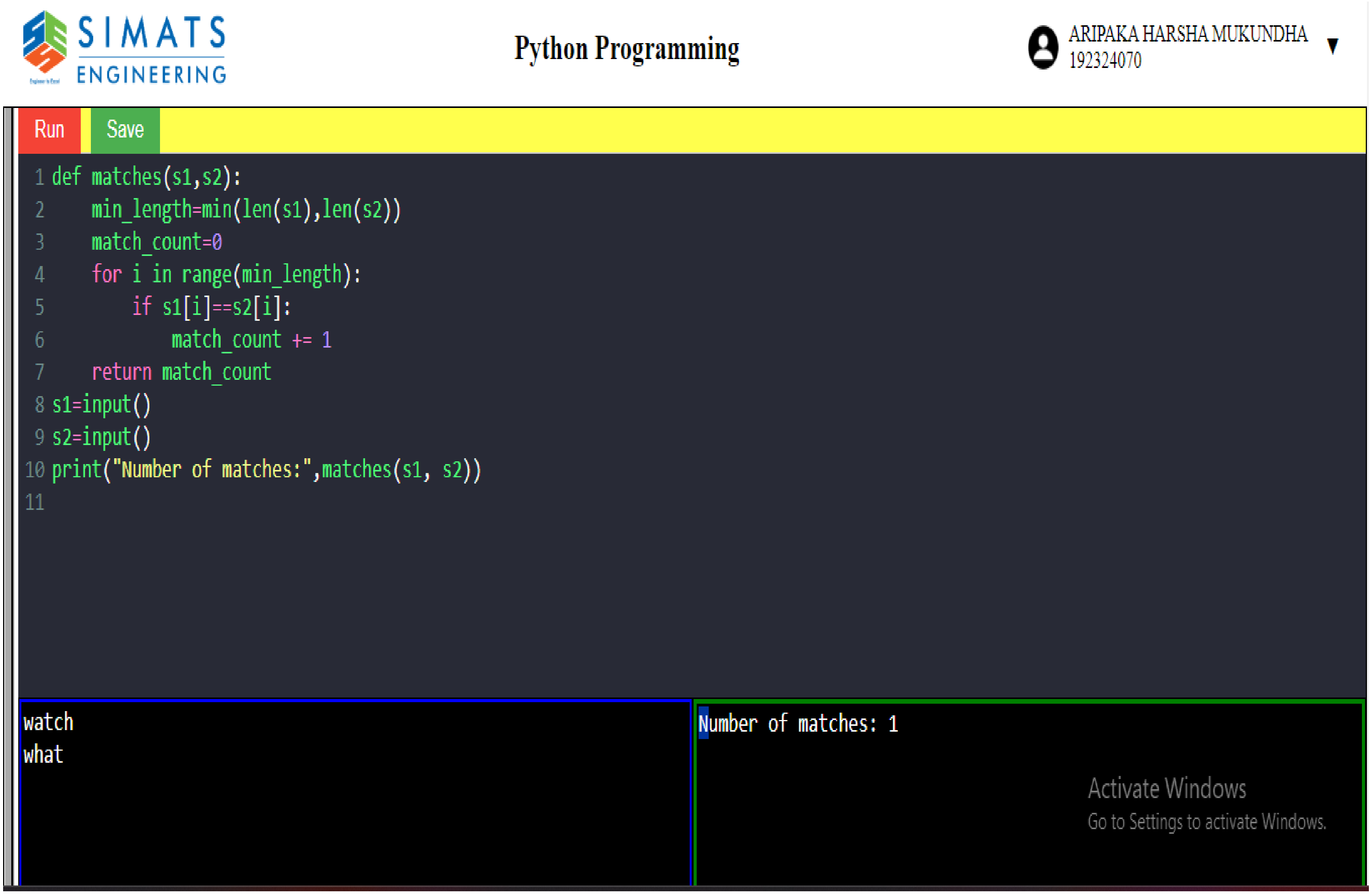


```
s=str(input())
print(s[::-1])
```

```
123456
```

```
654321
```

10)



```python
def matches(s1,s2):
    min_length=min(len(s1),len(s2))
    match_count=0
    for i in range(min_length):
        if s1[i]==s2[i]:
            match_count += 1
    return match_count
s1=input()
s2=input()
print("Number of matches:",matches(s1, s2))
```

```
watch
what
```

```
Number of matches: 1
```