****

Chandubhai S Patel Institute of Technology

Department of Computer Engineering

**IIT-Bombay E-Yantra Robotics Competition 2022-23**

*Task 0 Submission Documentation*

*Team Members:*

1. Pooja Ramani(22ce106)
2. Kaushal Bhanderi(22ce005)
3. Jalay Movaliya(22ce071)
4. Yash Nasit(22cs041)

*Task 0 Completion:*

1. Windows Installation:

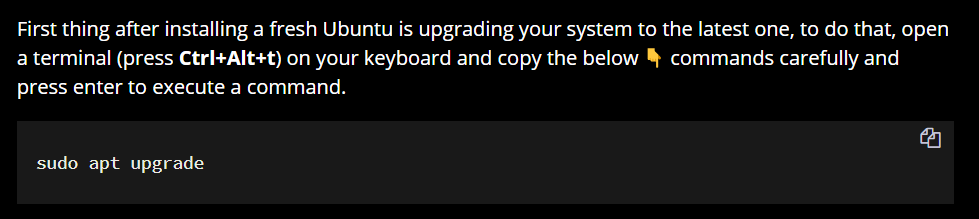
A screen shot of a computer screen

Description automatically generated

1. Installation of Ubuntu 22.04.3 and ROS2

[Ubuntu Download Link](https://ubuntu.com/download/desktop/thank-you?version=22.04.3&architecture=amd64)

ROS2 Installation:



A screenshot of a computer program

Description automatically generated

A black screen with white text

Description automatically generated

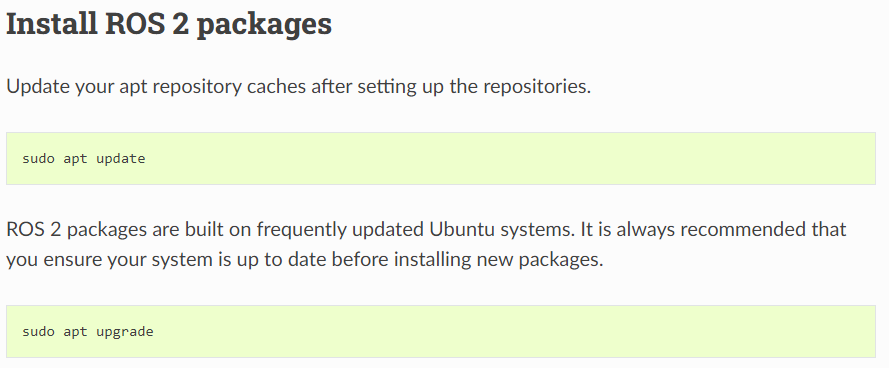
Now add the ROS 2 GPG key with apt.

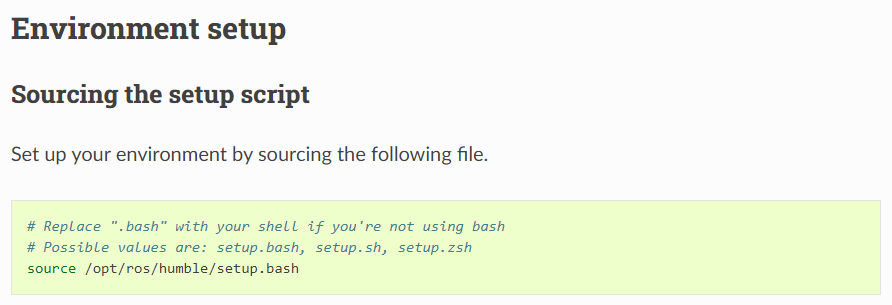
sudo apt update && sudo apt install curl -y

sudo curl -sSL [https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg](https://raw.githubusercontent.com/ros/rosdistro/master/ros.key%20-o%20/usr/share/keyrings/ros-archive-keyring.gpg)

Then add the repository to your sources list.

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null





A screenshot of a computer program

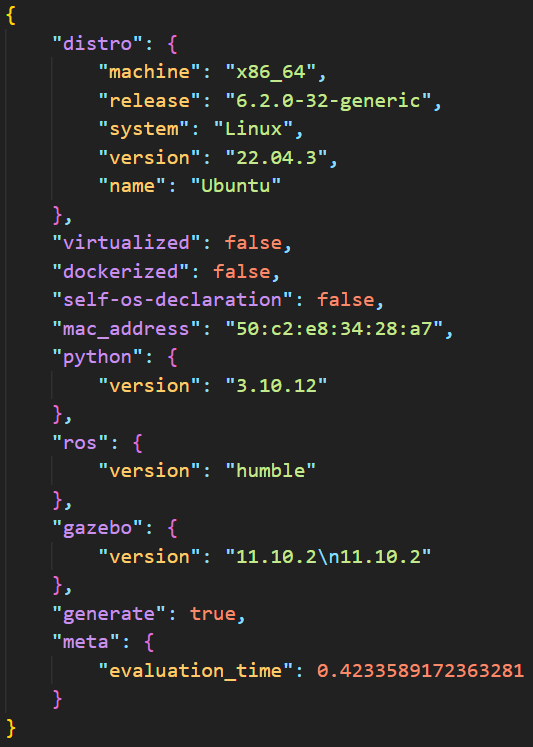
Description automatically generated

Gazebo Installation:

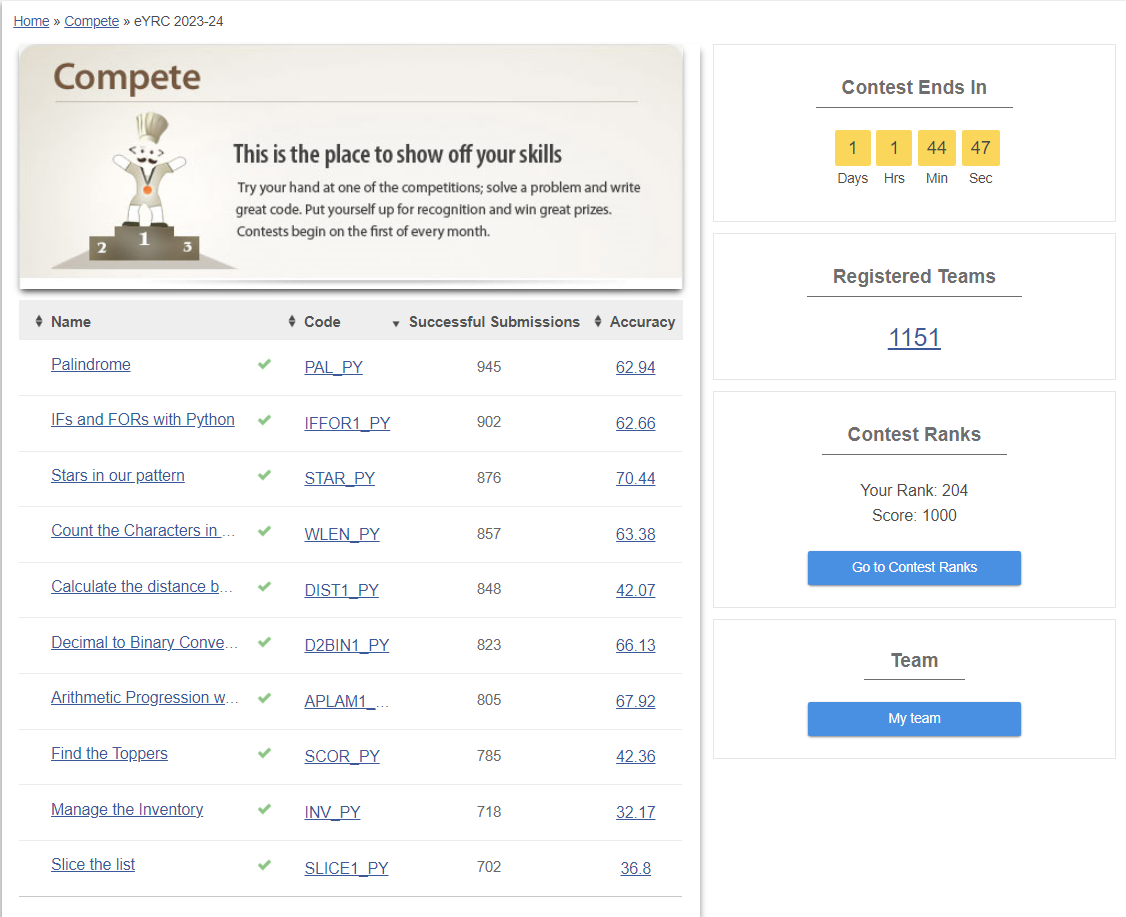
A screenshot of a computer

Description automatically generated

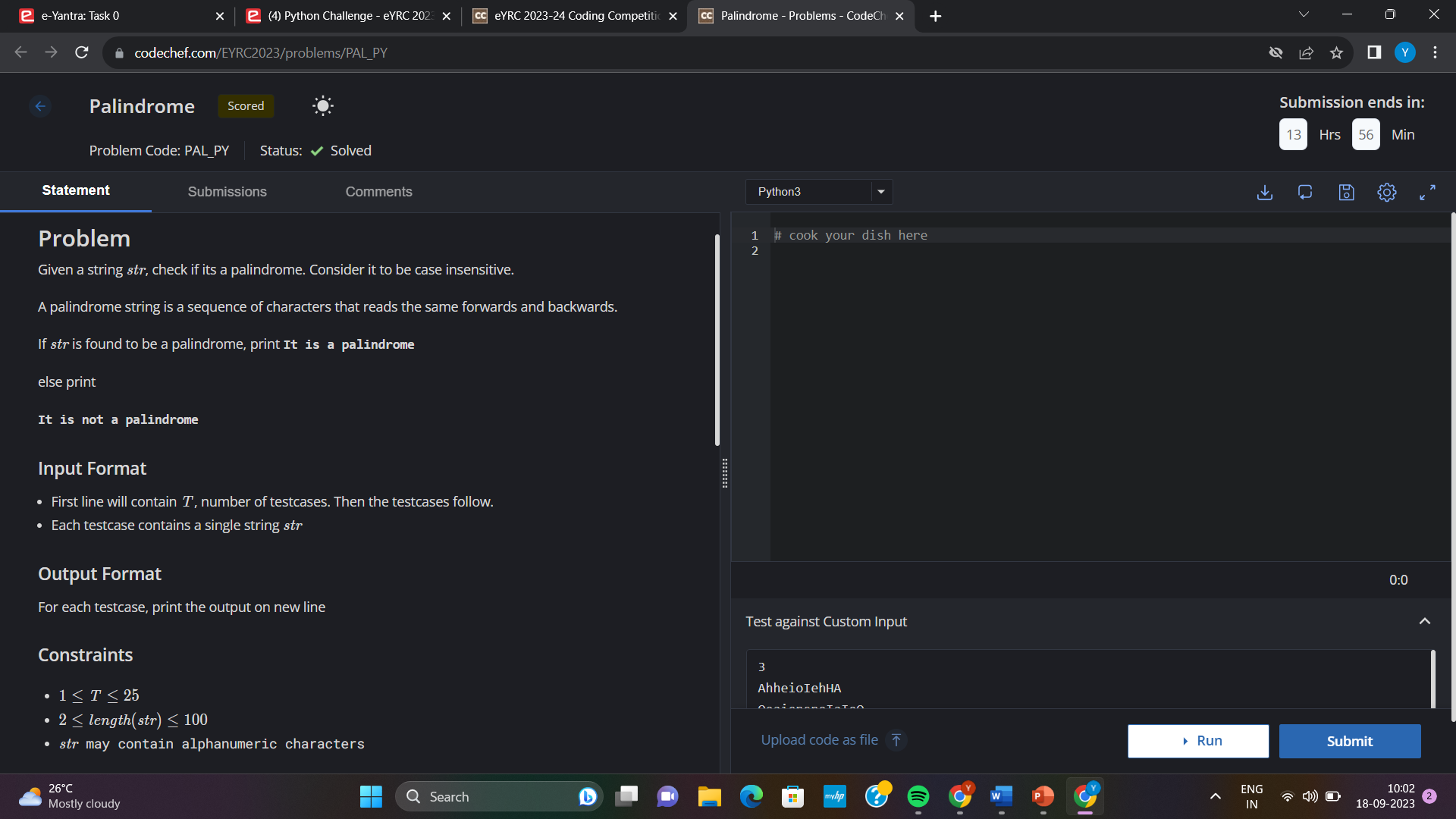
Generated JSON file:



1. Python Challenges:



Python Codes:

*1)*



*Code:*

t= int(input())

while(t!=0):

s= input()

s=s.upper()

rs=s[: :-1]

if(s==rs):

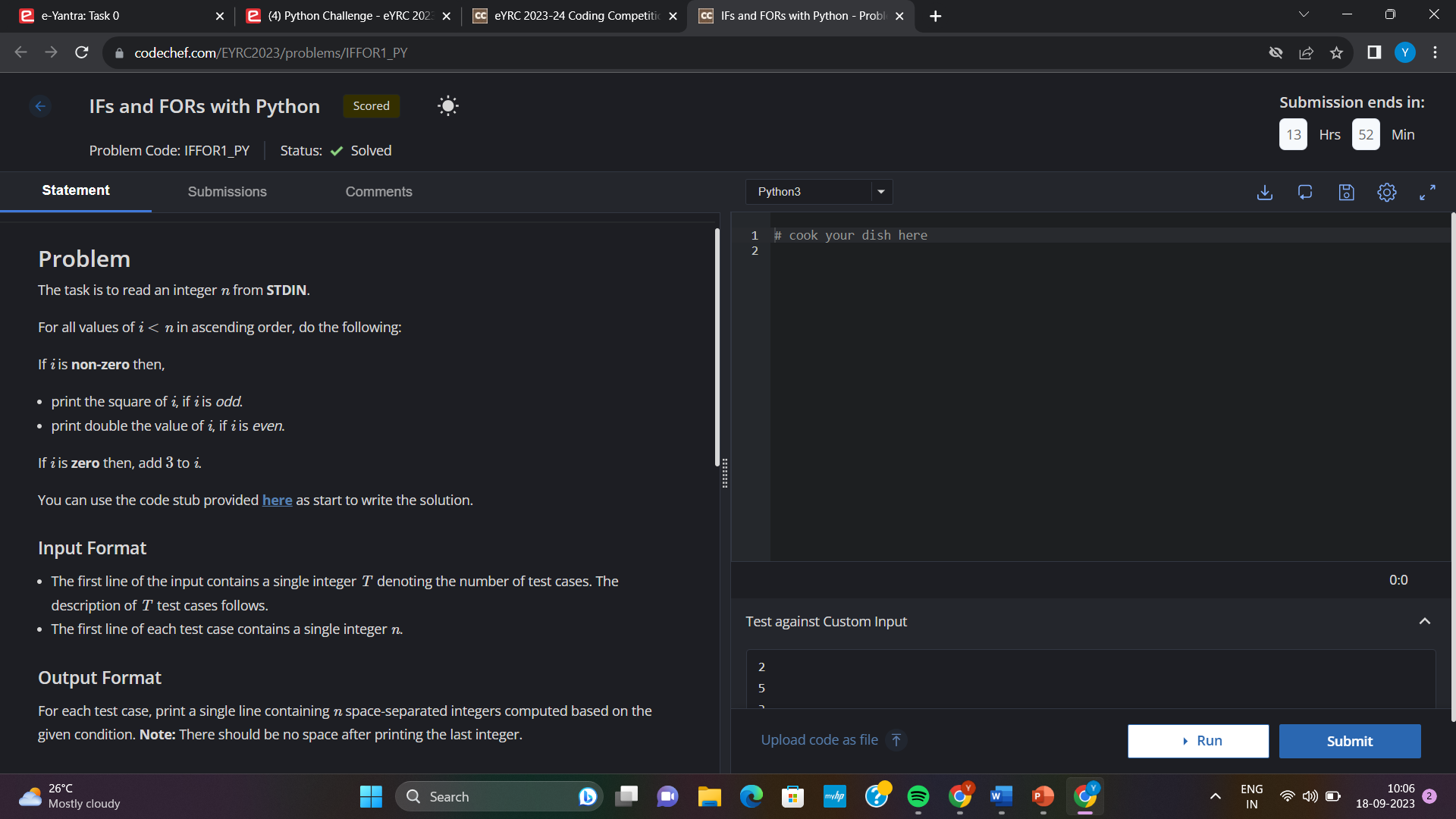
print("It is a palindrome")

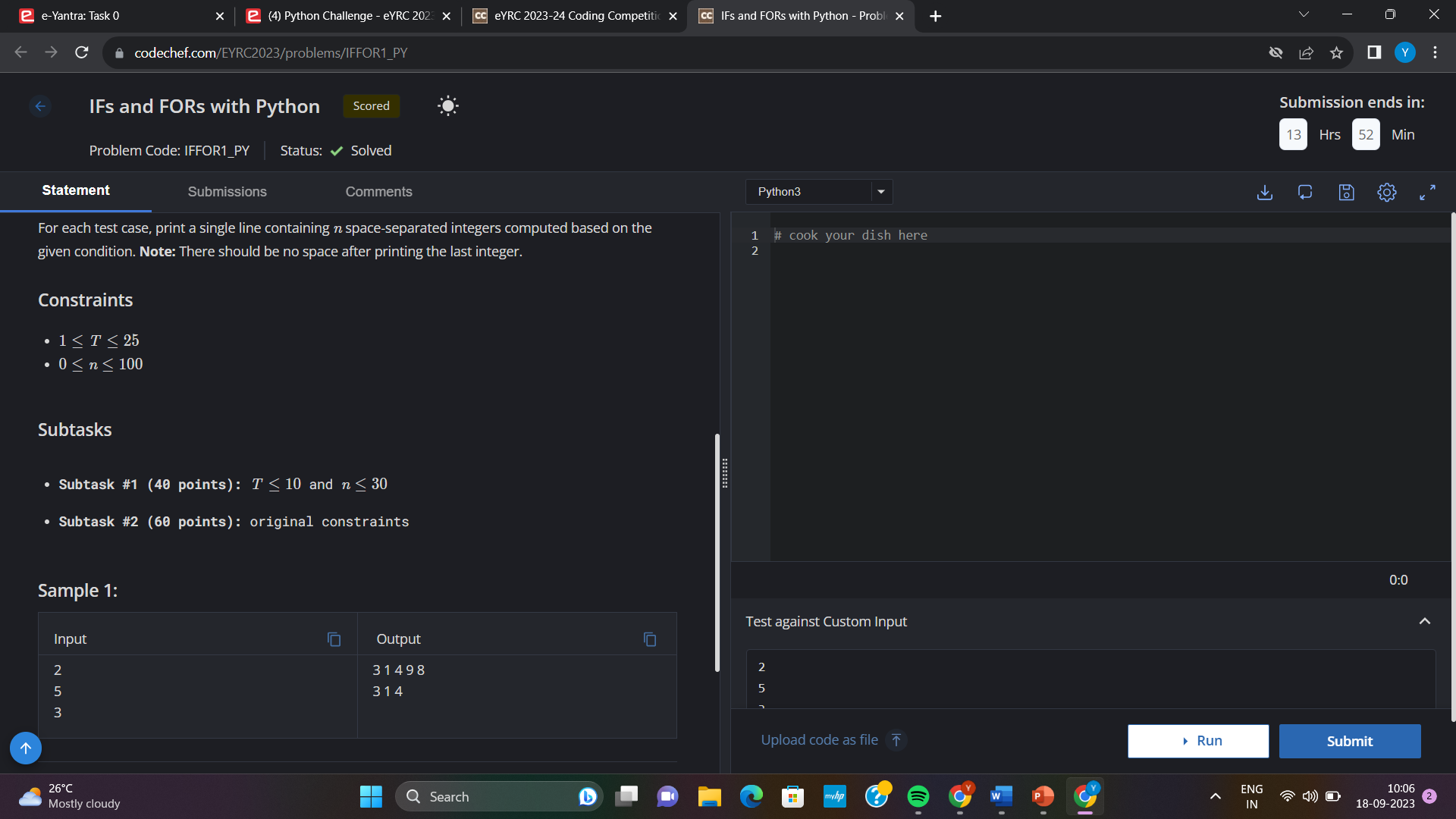
else:

print("It is not a palindrome")

t-=1

*2)*





*Code:*

t= int(input())

while (t!=0):

n= int(input())

s=""

for i in range (0,n,1):

if(i==0):

s=s+'3'+" "

else:

if(i%2==0):

s=s+str(2\*i)+" "

else:

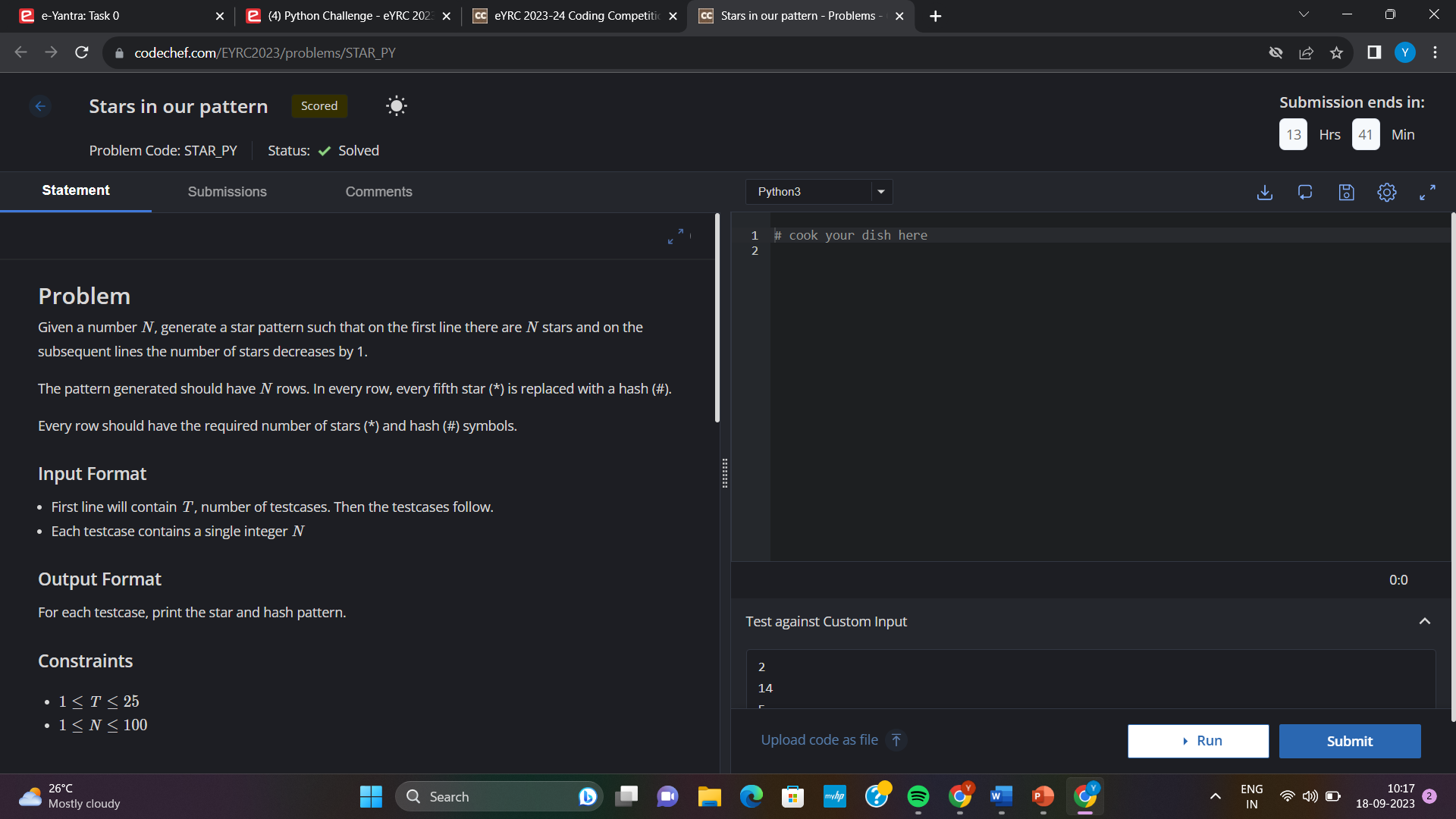
s=s+str(i\*i)+" "

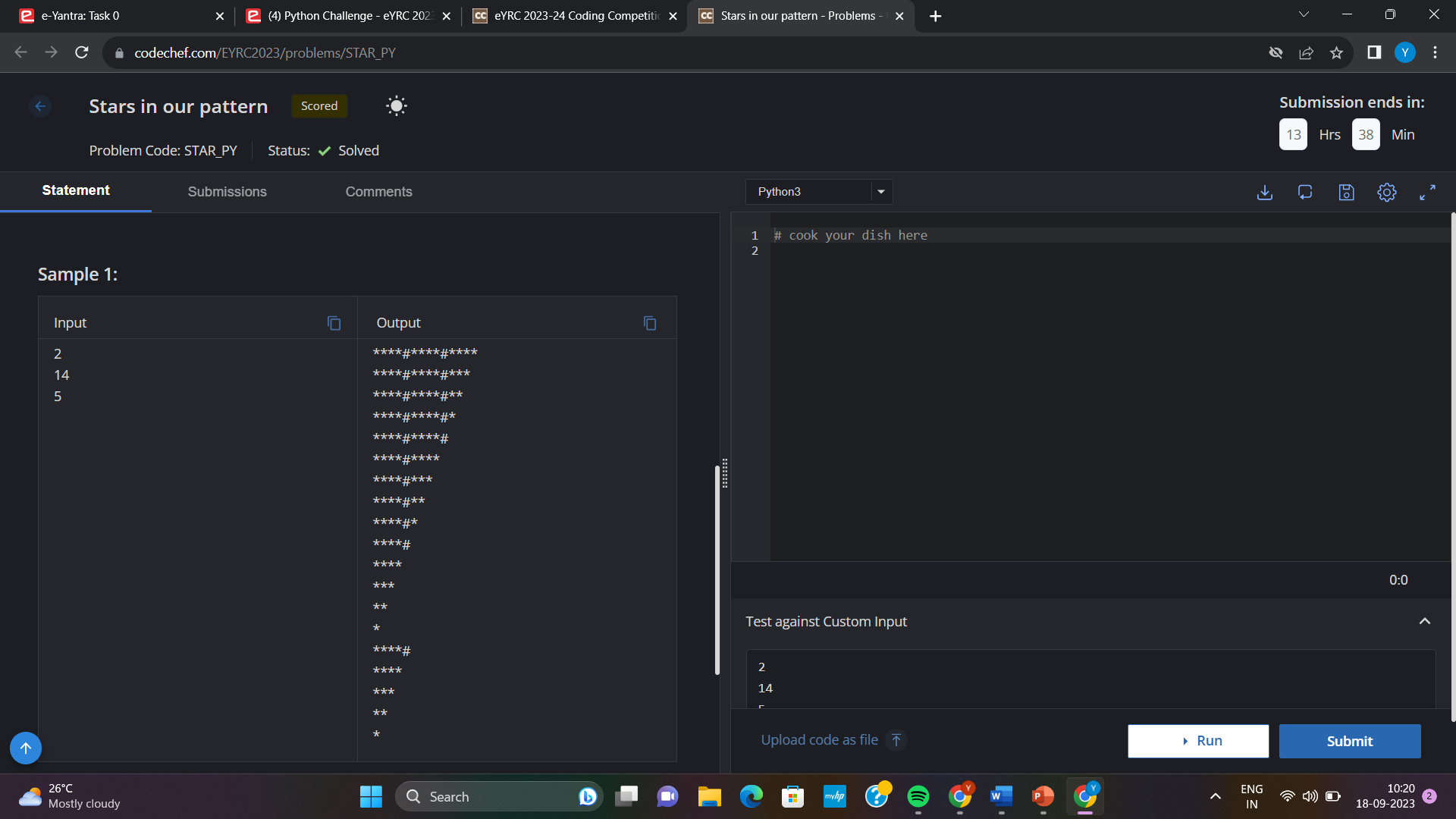
s.strip()

print(s)

t-=1

*3)*

**

**

*Code:*

t=int(input())

while(t!=0):

n=int(input())

for i in range(n,0,-1):

for j in range(1,i+1):

if(j%5==0):

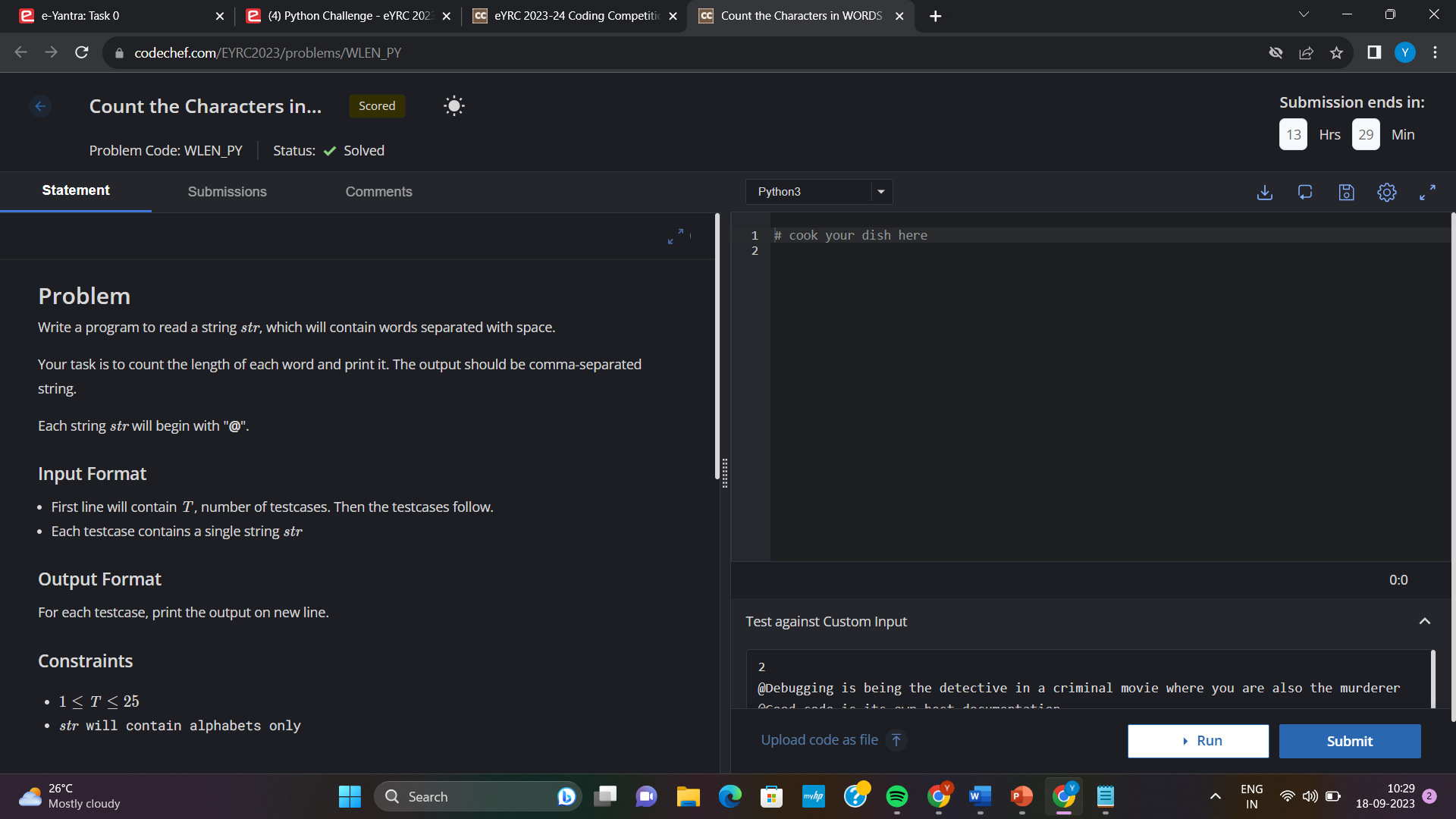
print("#",end="")

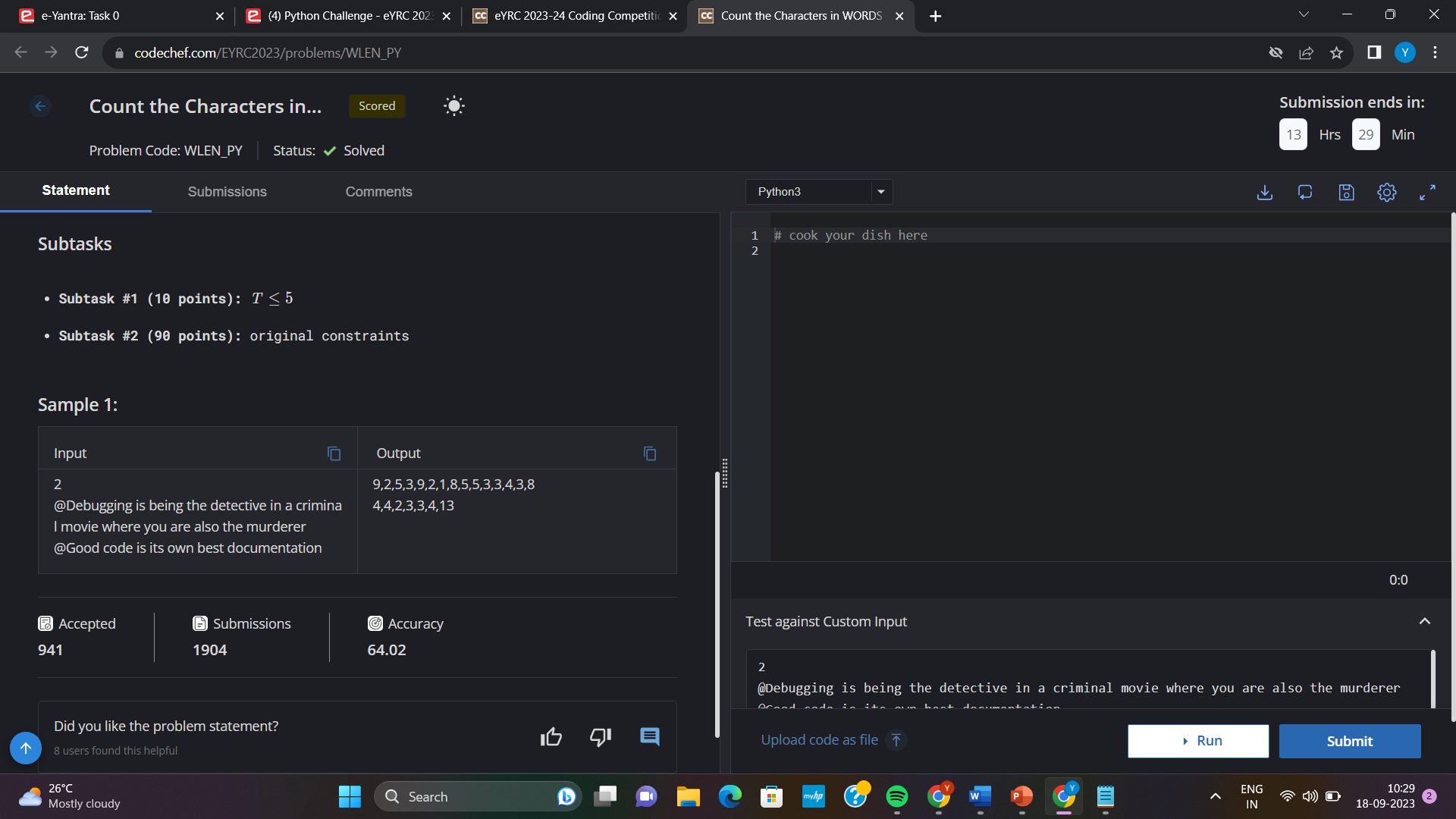
else:

print("\*",end="")

print()

t-=1

*4)*



*Code:*

t=int(input())

while(t!=0):

s=input().split(" ")

counts=""

for i in s:

c=0

for j in i:

if(j=='@'):

continue

else:

c+=1

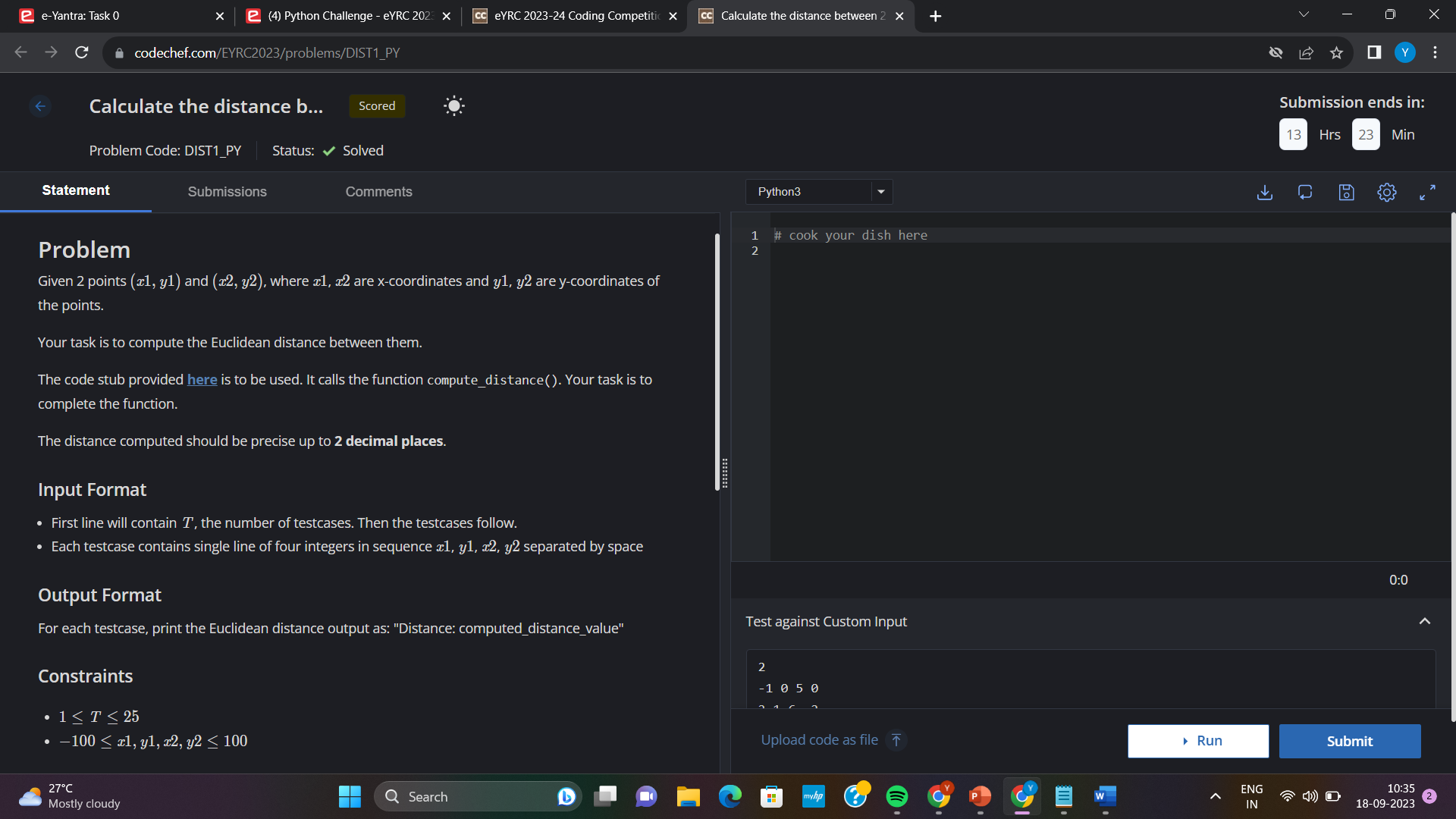
counts+= str(c)

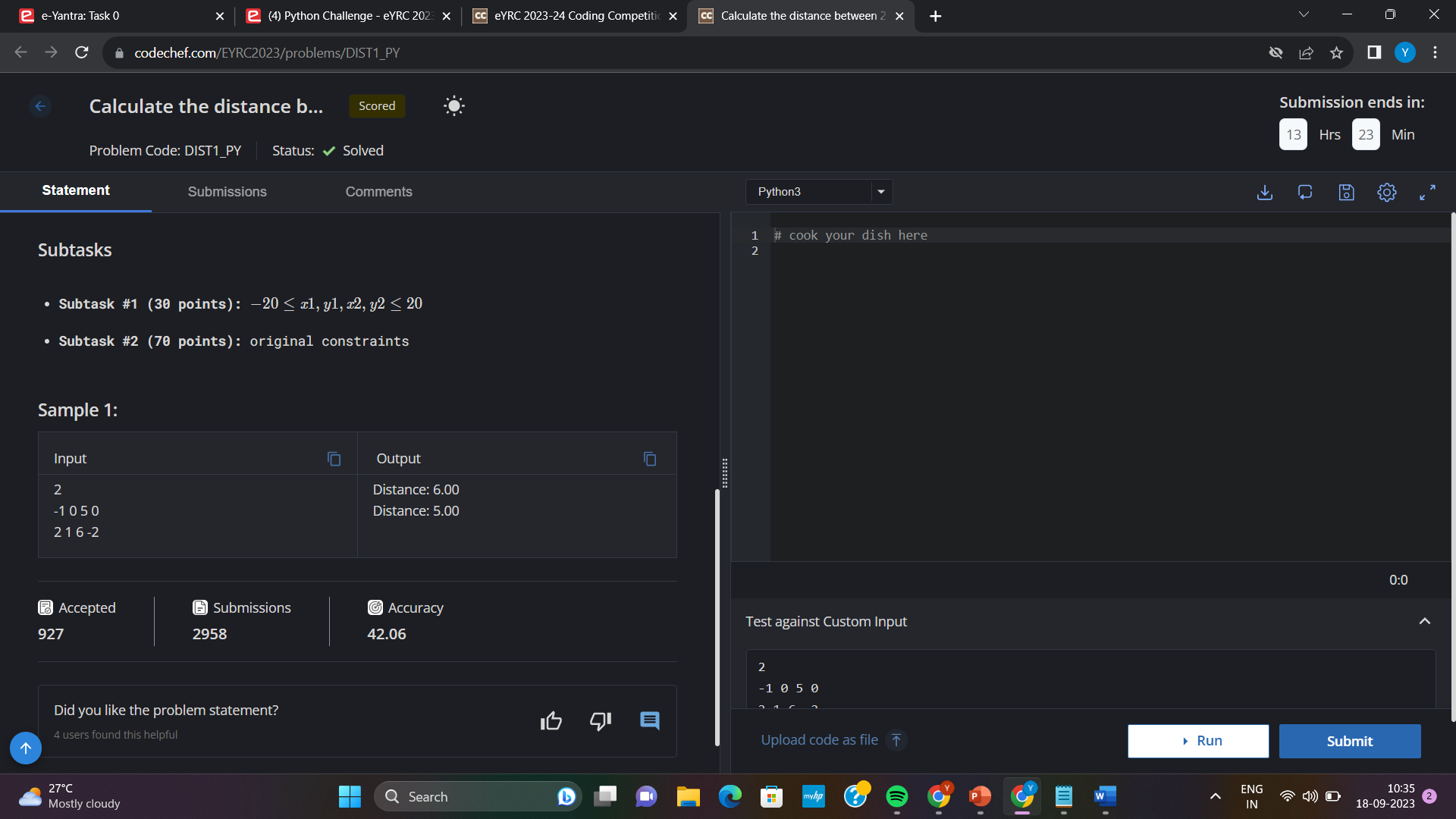
counts+=","

print(counts[:len(counts)-1])

t-=1

*5)*

**

**

*Code:*

import math

# Import any required module/s

# Function to calculate Euclidean distance between two points

def compute\_distance(x1, y1, x2, y2):

distance = math.sqrt((x2-x1)\*\*2 + (y2-y1)\*\*2)

distance= round(distance,2)

print("Distance: "+str(distance))

# Complete this function to return Euclidean distance and

# print the distance value with precision up to 2 decimal places

# Main function

if \_\_name\_\_ == '\_\_main\_\_':

# Take the T (test\_cases) input

test\_cases = int(input())

while(test\_cases!=0):

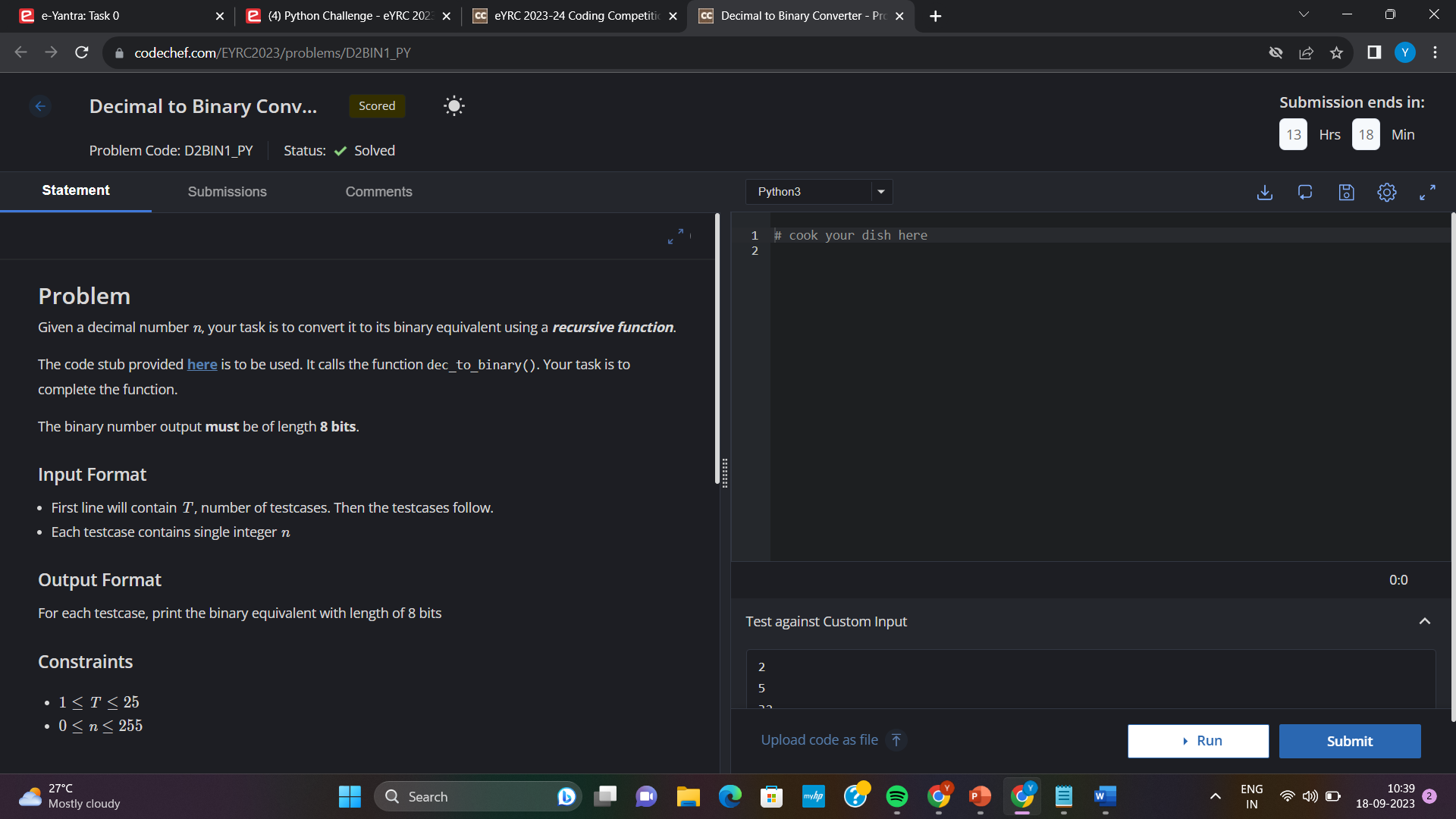
# Write the code here to take the x1, y1, x2 and y2 values

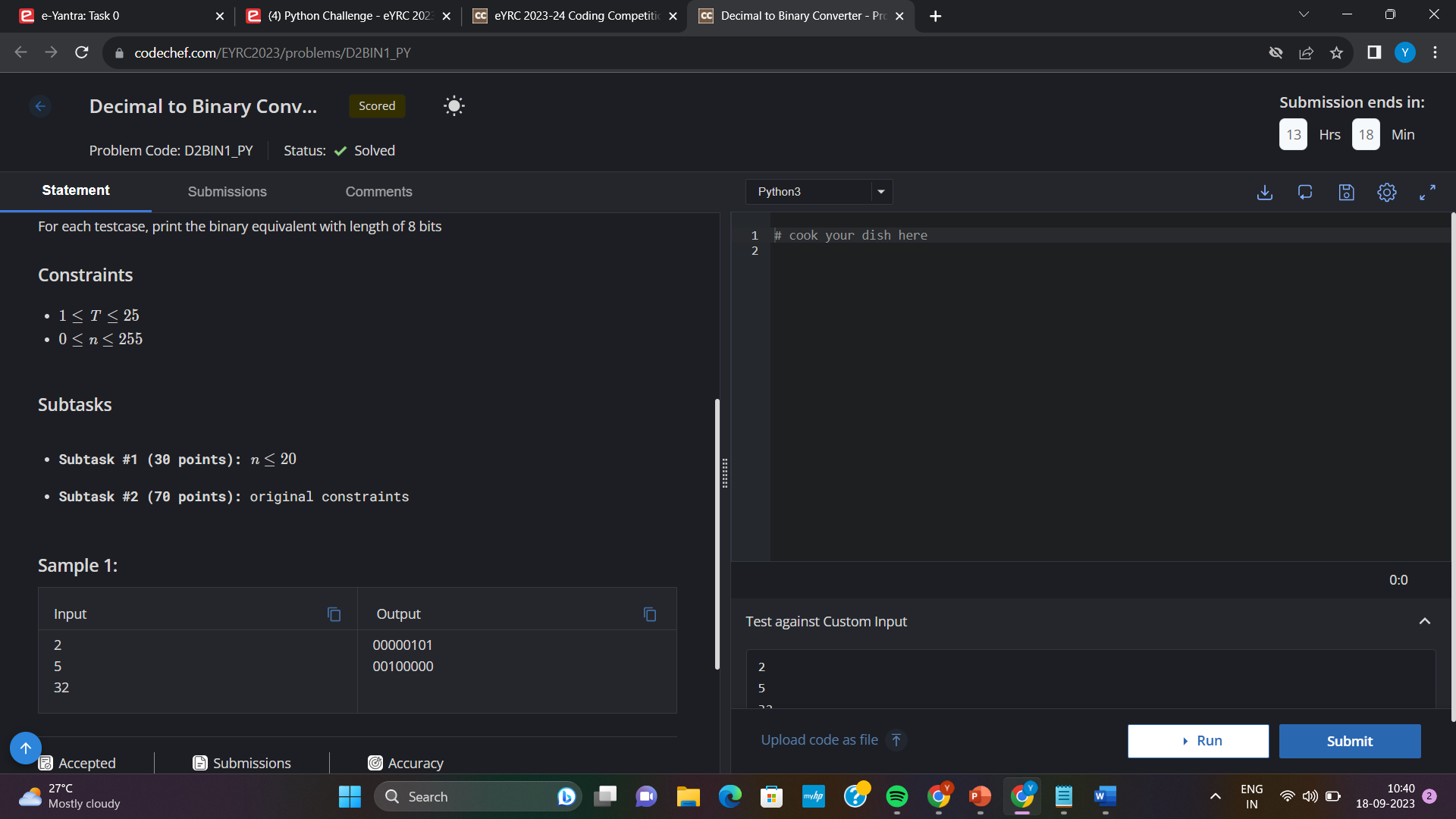
x1,y1,x2,y2= input().split(" ")

# Once you have all 4 values, call the compute\_distance function to find Euclidean distance

compute\_distance(int(x1), int(y1), int(x2), int(y2))

test\_cases-=1

*6)*

**

*Code:*

b=""

while(n!=0):

b=b+str(n%2)

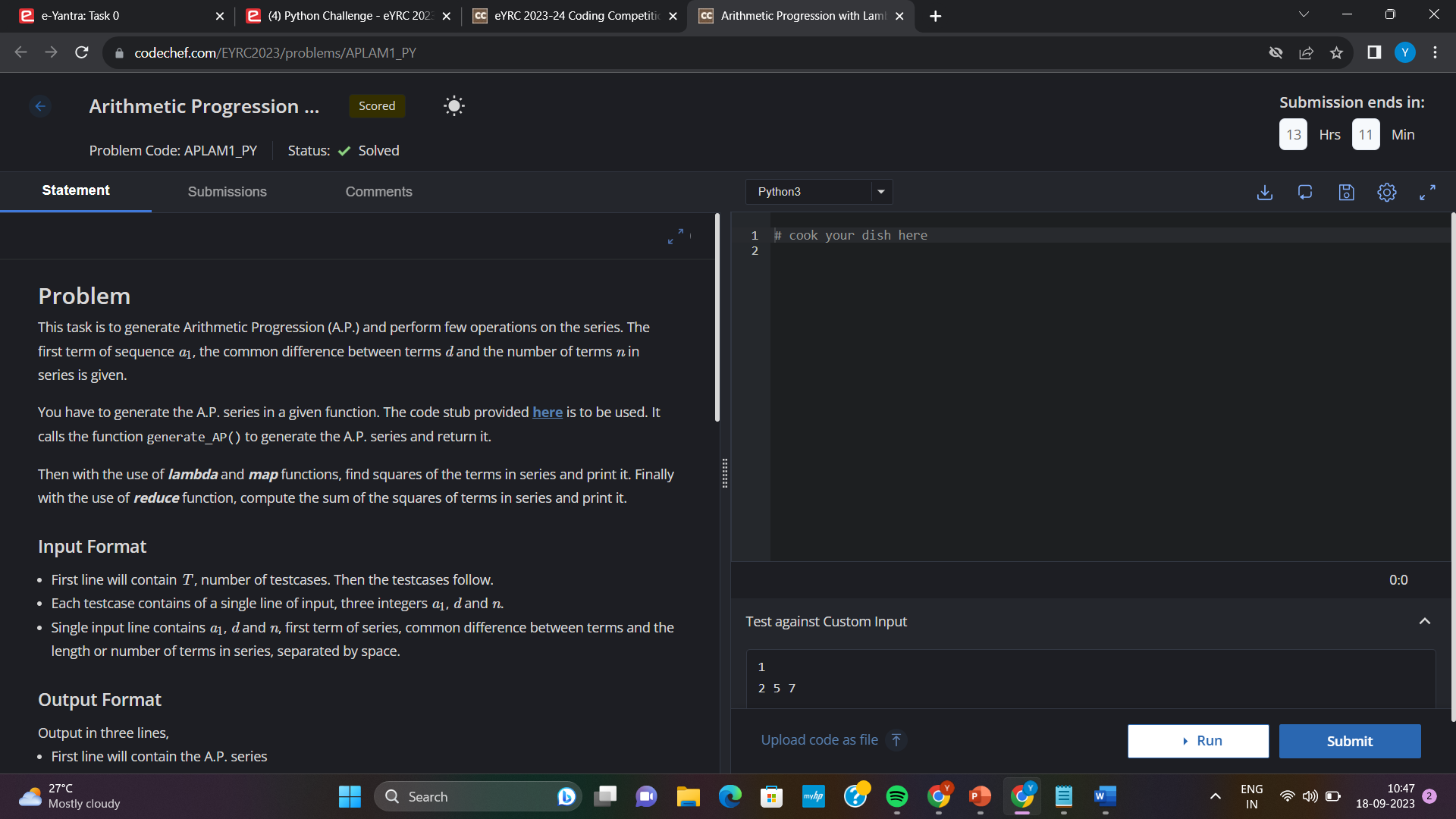
n=n/2

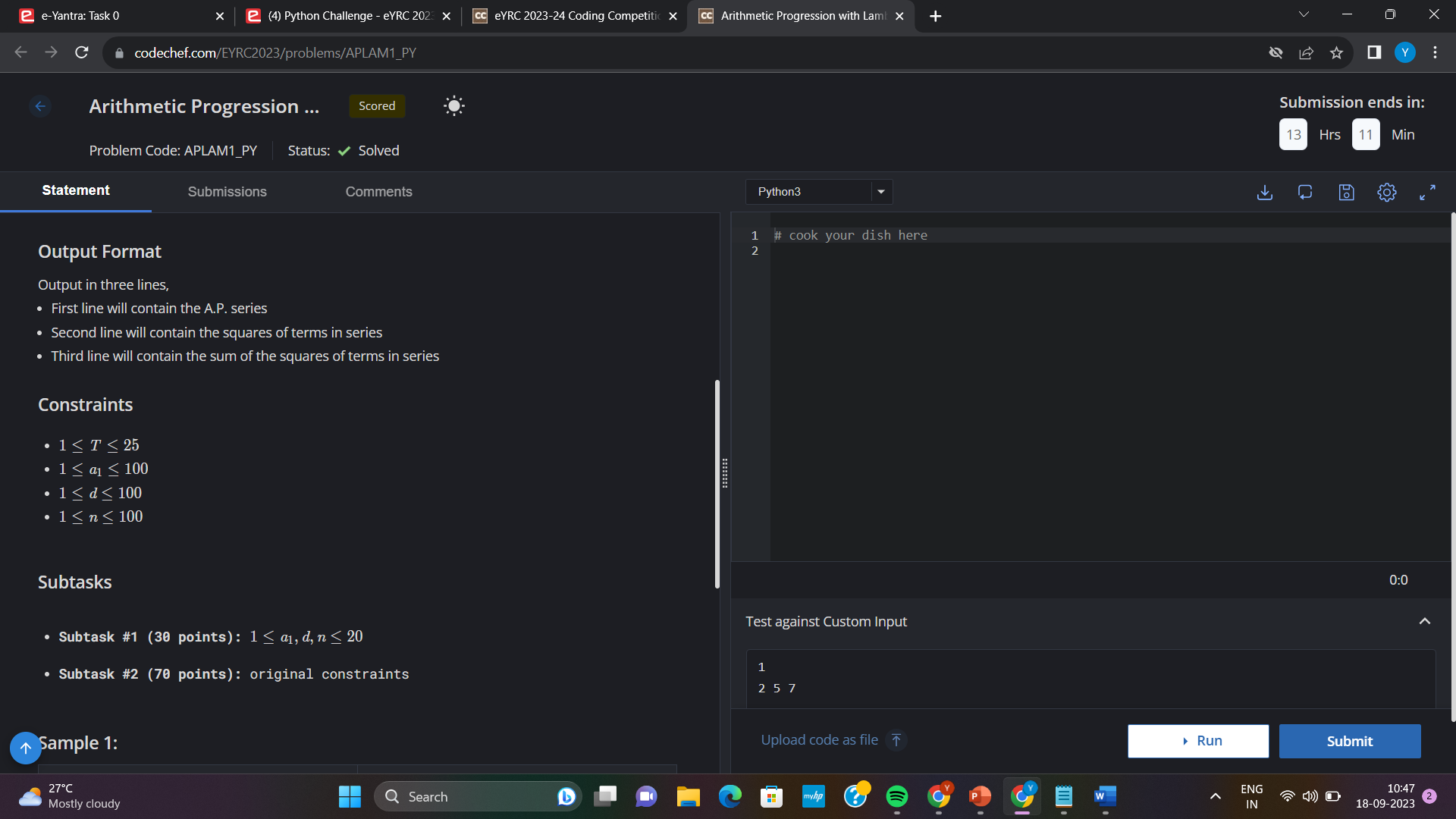
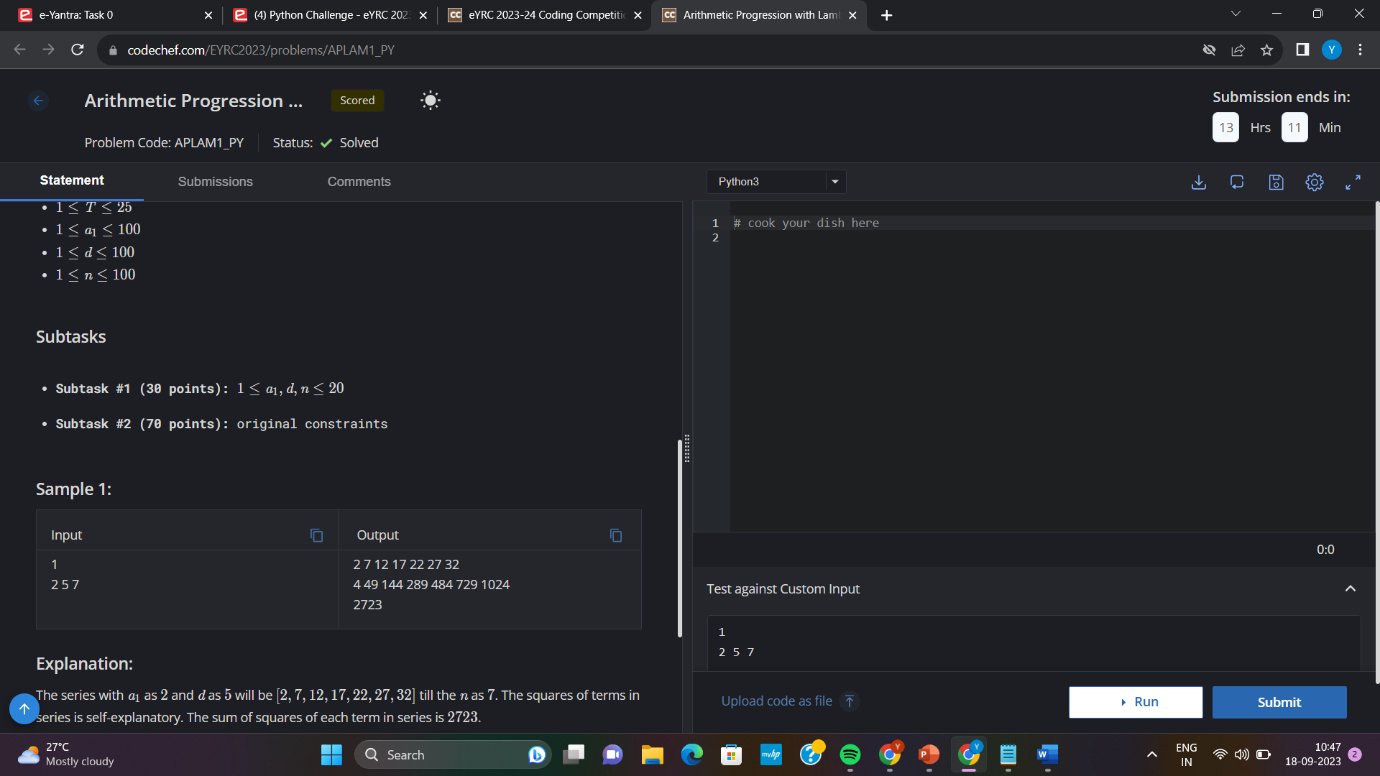
while(len(b)<8):

b+="0"

bin\_num = b[:-1]

*7)*

**

**

*Code:*

from functools import reduce

t = int(input())

def series(a, d, n):

ap = [a + (i \* d) for i in range(n)]

return ap

for i in range(t):

str = input().split()

a = int(str[0])

d = int(str[1])

n = int(str[2])

s = series(a, d, n)

for j in s:

print(j, end=" ")

print()

sq = list(map(lambda x: x\*\*2, s))

for j in sq:

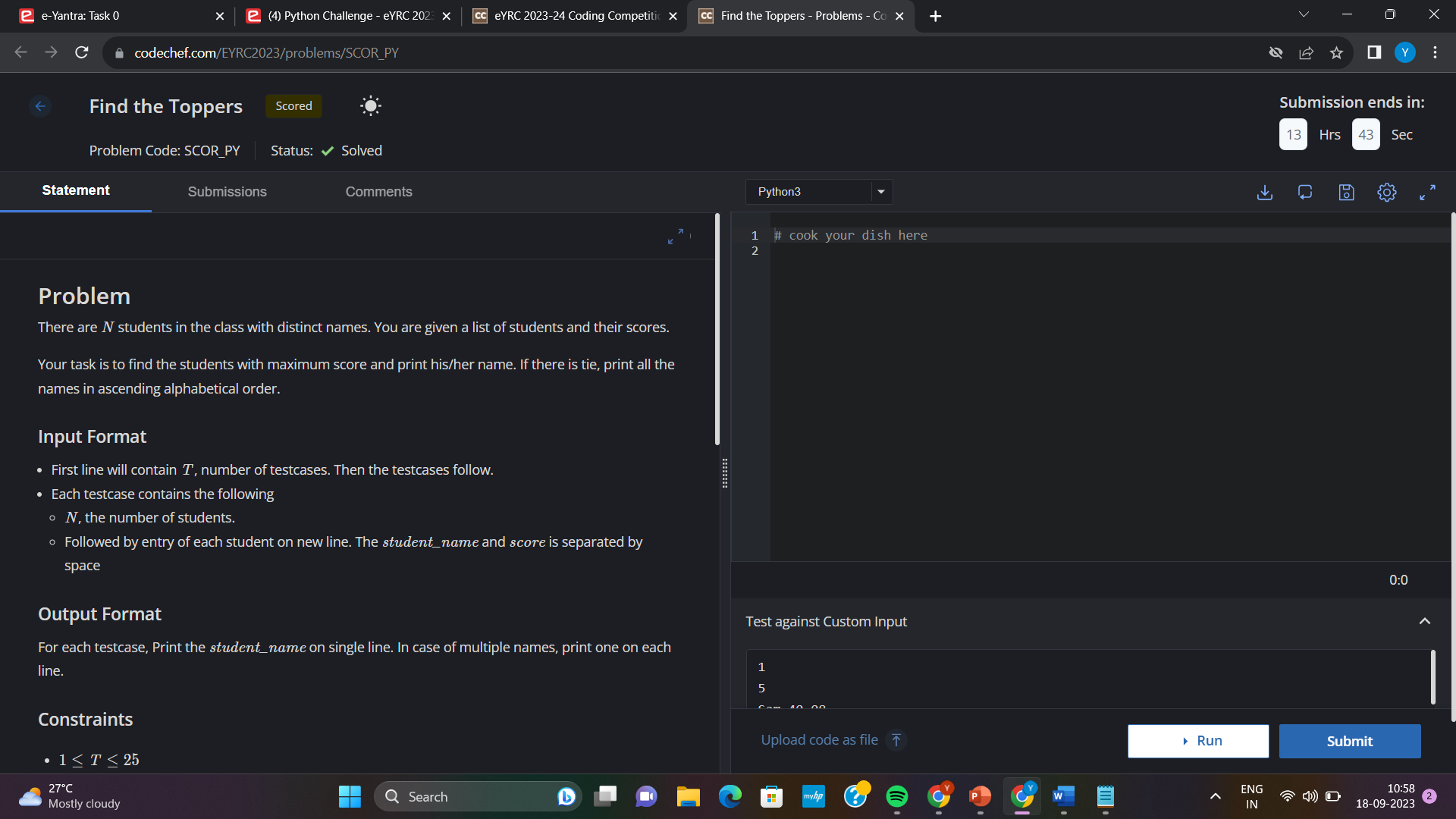
print(j, end=" ")

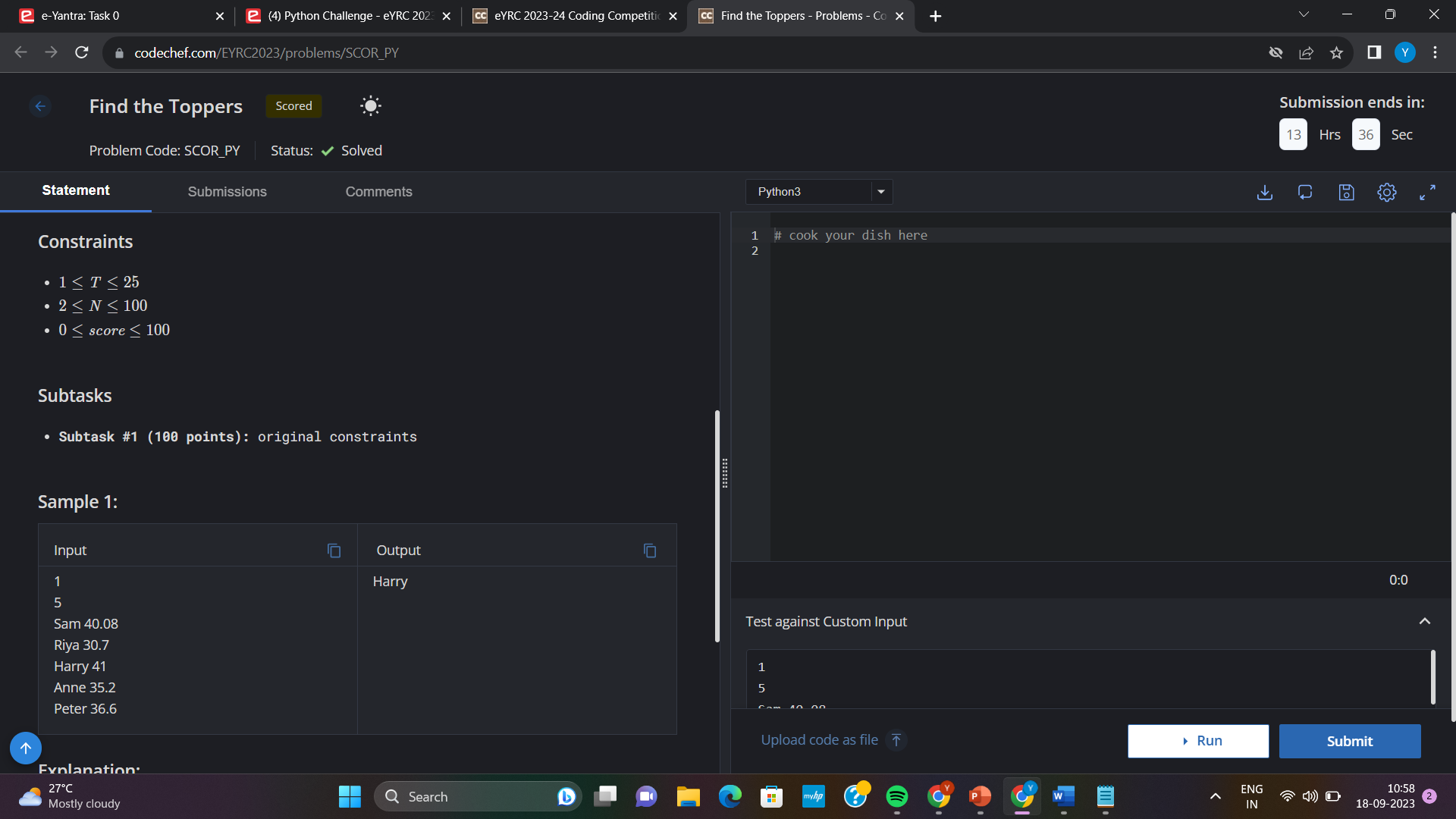
print()

Sum = reduce(lambda x, y: x + y, sq)

print(Sum)

*8)*

**

**

*Code:*

T = int(input(""))

for i in range(T):

n = int(input(""))

names = []

marks = []

str = ""

for j in range(n):

str = input("").split()

names.append(str[0].lower())

marks.append(float(str[1]))

ans = 0

toppers = []

maxmarks = -1

for j in range(n):

if marks[j] > maxmarks:

maxmarks = marks[j]

toppers = [names[j]]

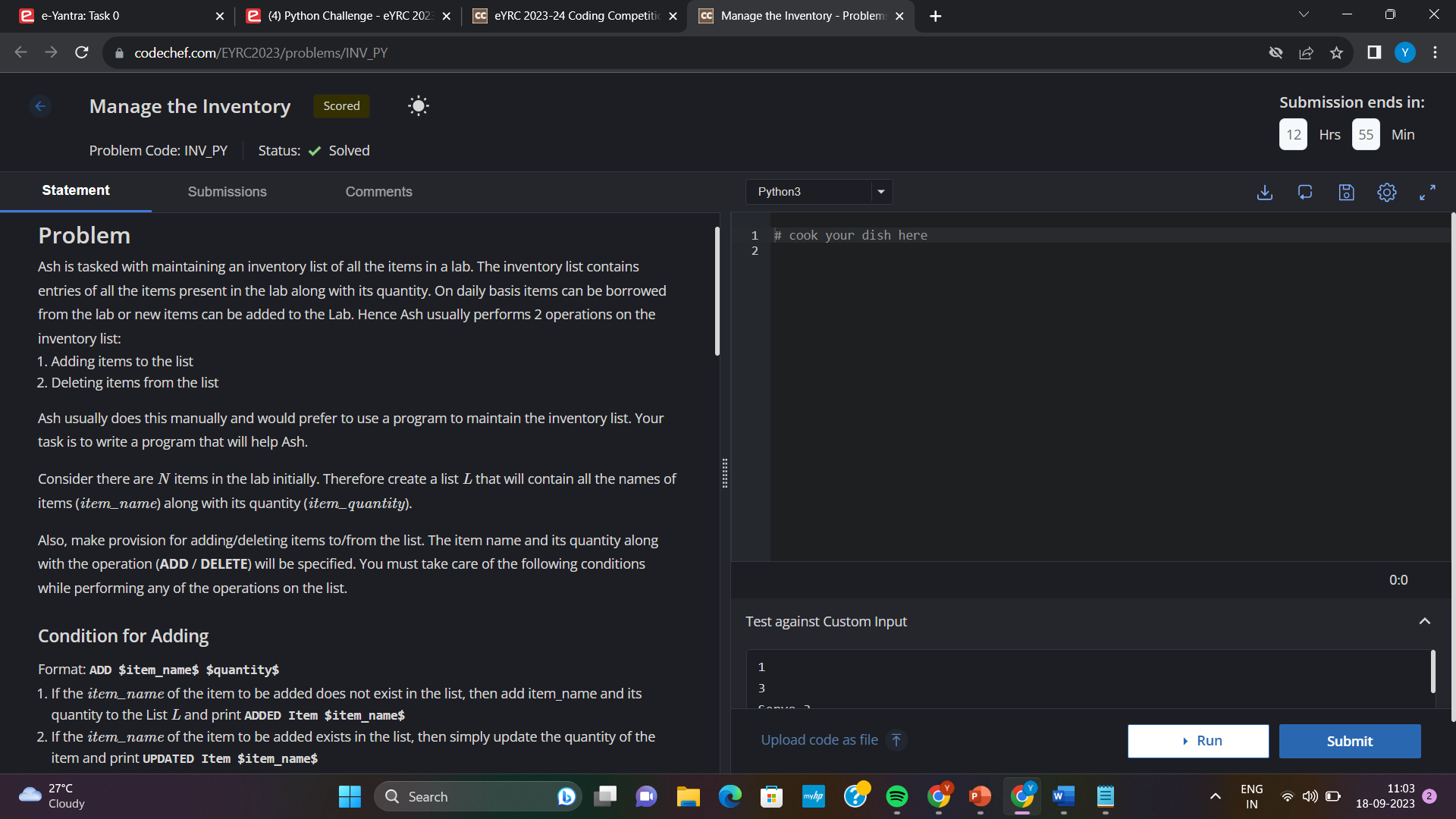
elif marks[j] == maxmarks:

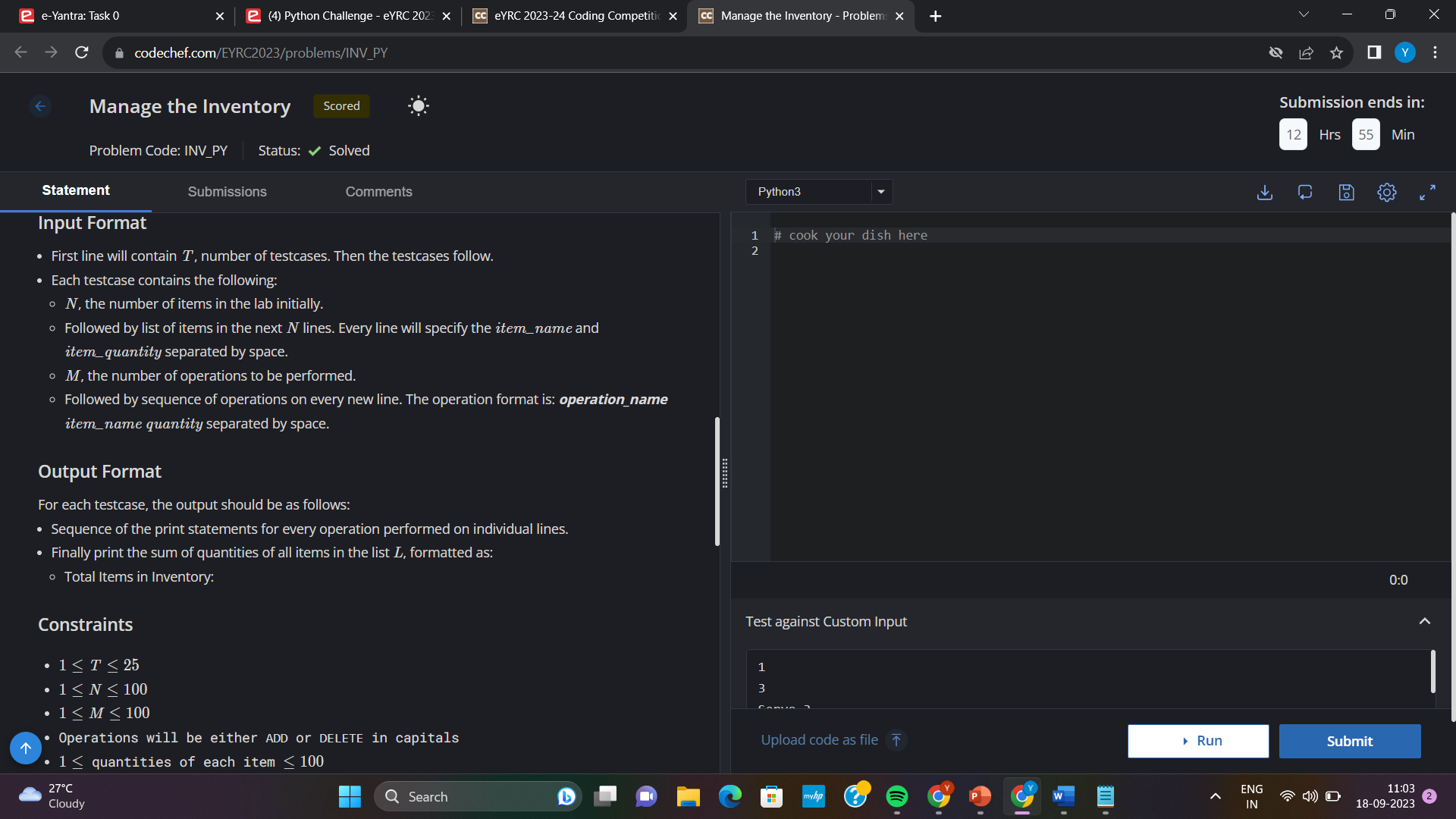
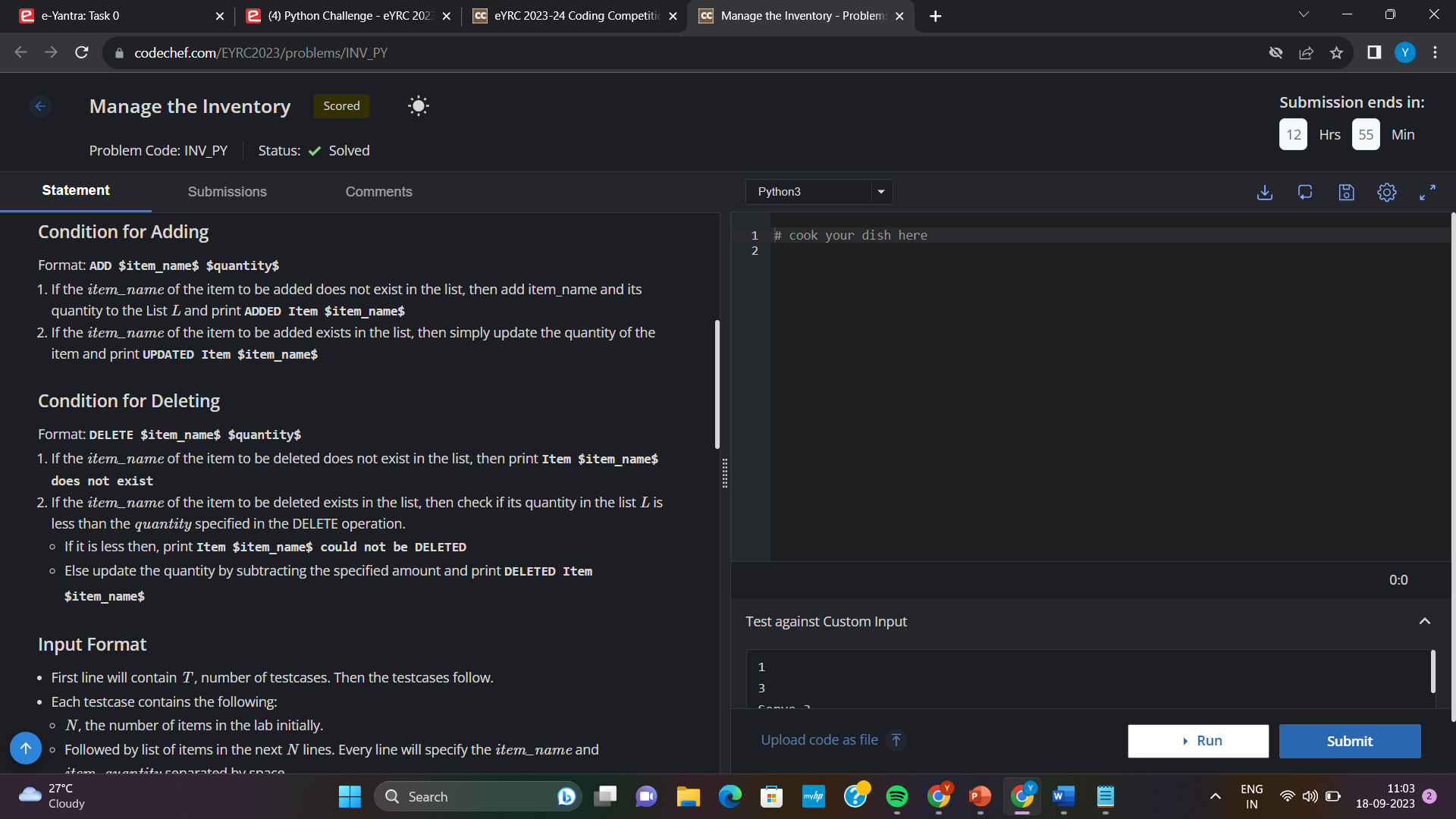
toppers.append(names[j])

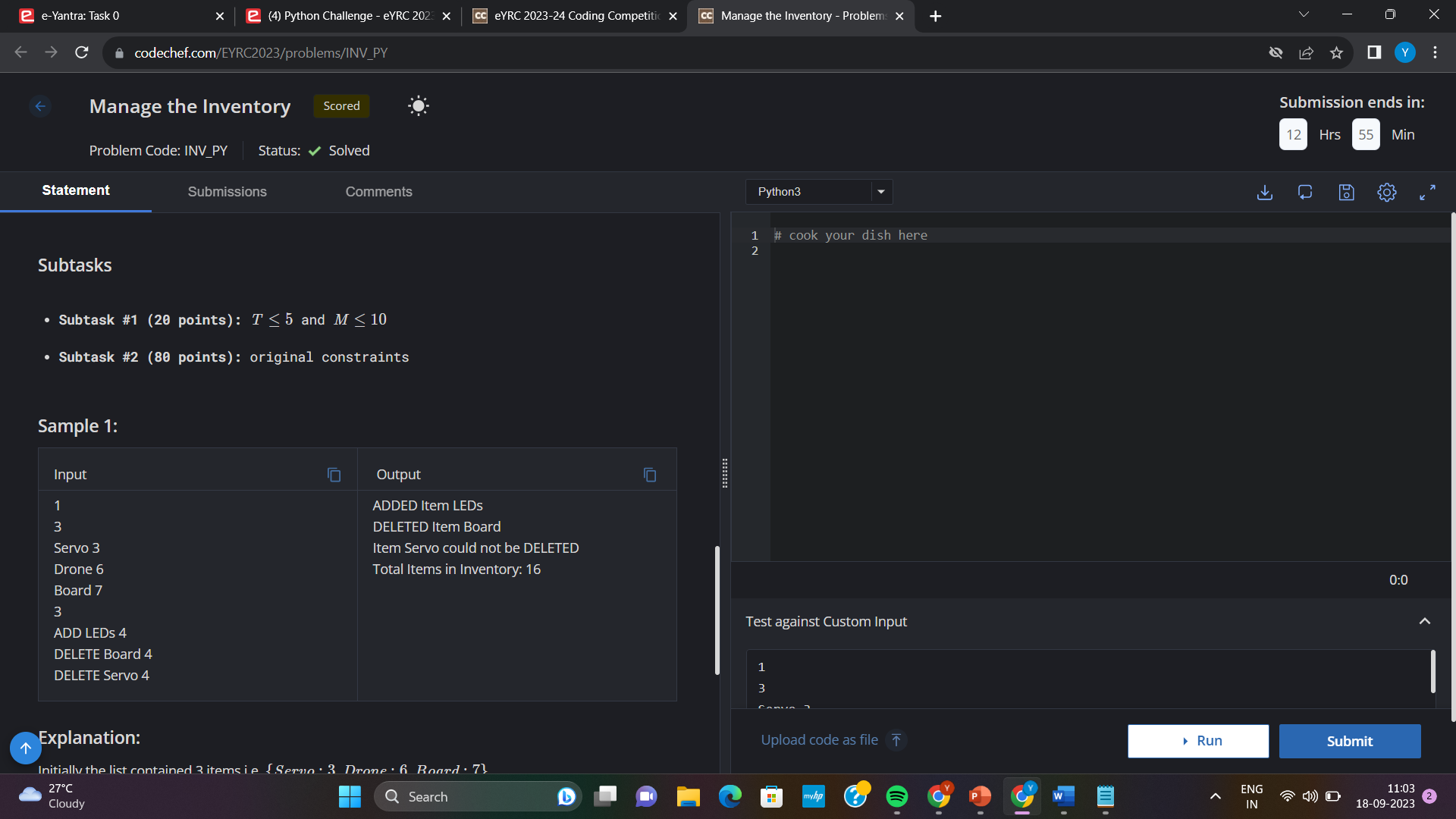
toppers.sort()

for topper in toppers:

print(topper[0].upper() + topper[1:len(topper)])

* 8)*

**

**

*Code:*

t = int(input())

for i in range(t):

n = int(input())

item = {}

for j in range(n):

s = input().split()

item[s[0]] = int(s[1])

m = int(input())

for j in range(m):

s = input().split()

com = s[0]

name = s[1]

quantity = int(s[2])

if(com == "ADD"):

if item.get(name):

item[name]+=quantity

print("UPDATED Item " + name)

else:

item[name] = quantity

print("ADDED Item " + name)

n+=1

elif(com == "DELETE"):

if item.get(name):

if(item[name] >= quantity):

item[name]-=quantity

print(f"DELETED Item {name}")

else:

print(f"Item {name} could not be DELETED")

else:

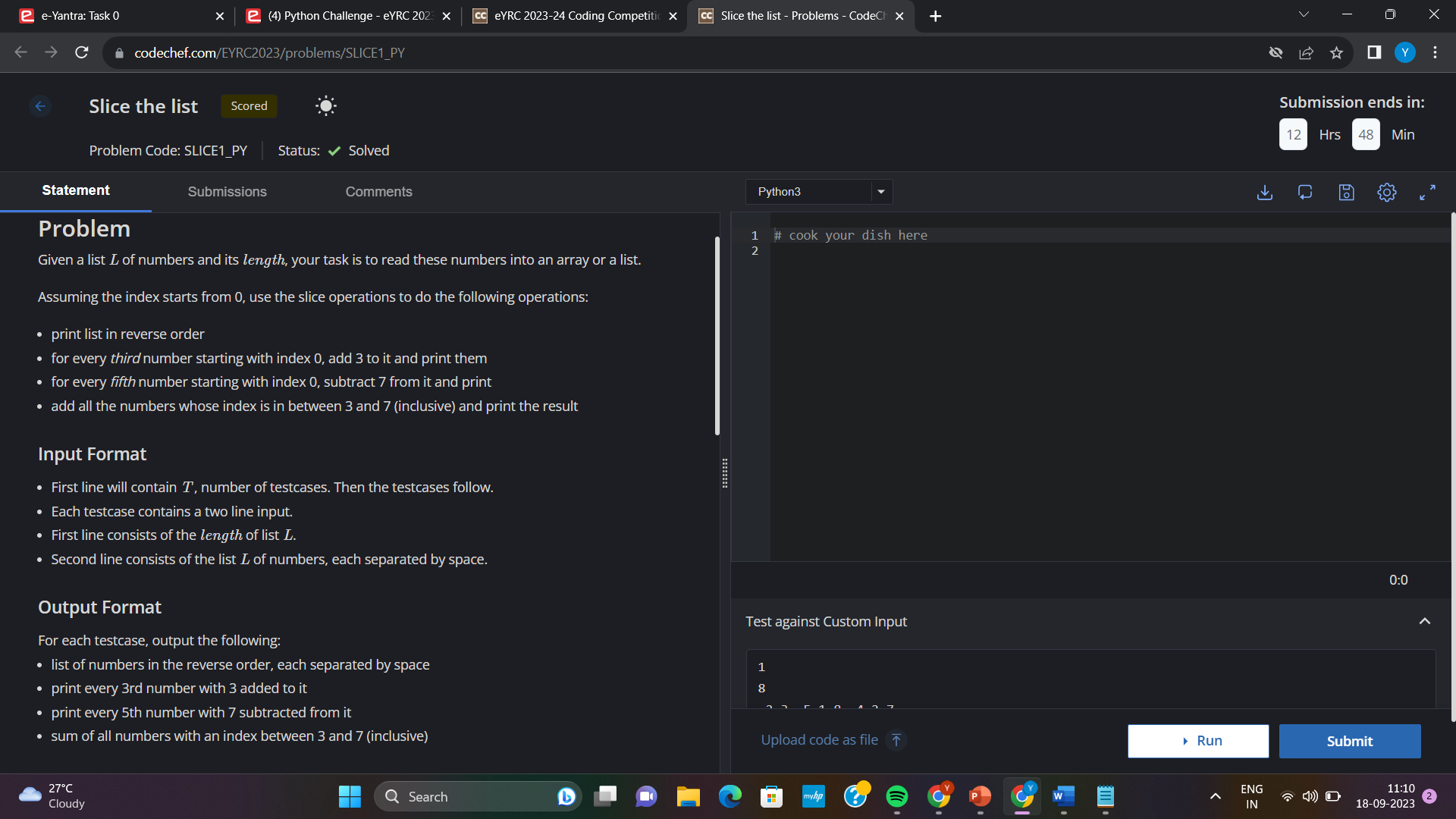
print(f"Item {name} does not exist")

s = 0

for key in item.values():

s += key

print(f"Total Items in Inventory: {s}")

*10)*



*Code:*

t = int(input())

for i in range(t):

n = int(input())

arr = input().split()

for j in range(n):

arr[j] = int(arr[j])

for j in range(n):

print(arr[n-j-1], end=" ")

print("\n", end="")

for j in range(1, n):

if j % 3 == 0:

print(str(arr[j]+3), end=" ")

print("\n", end="")

for j in range(1, n):

if j % 5 == 0:

print(arr[j]-7, end=" ")

print("\n", end="")

s = 0

if n > 4:

for j in range(3, 8):

s+=arr[j]

print(s)