**Introduction to programming-2 (Python)**

***#Amanbay Almas, 2215, 221105***

**Assignment 5 (parts: A&B)**

**Problem – A1**

**Solution code:**

**#ID 221105, Name Almas Amanbay, Group 2215**

**from pyfiglet import Figlet**

**def main():**

**font\_name, text = get\_input()**

**print\_text(font\_name, text)**

**def print\_text(font\_name, text):**

**f = Figlet(font = font\_name)**

**print(f.renderText(text))**

**def get\_input():**

**while True:**

**try:**

**font\_name = input()**

**text = input()**

**if text == "":**

**raise ValueError()**

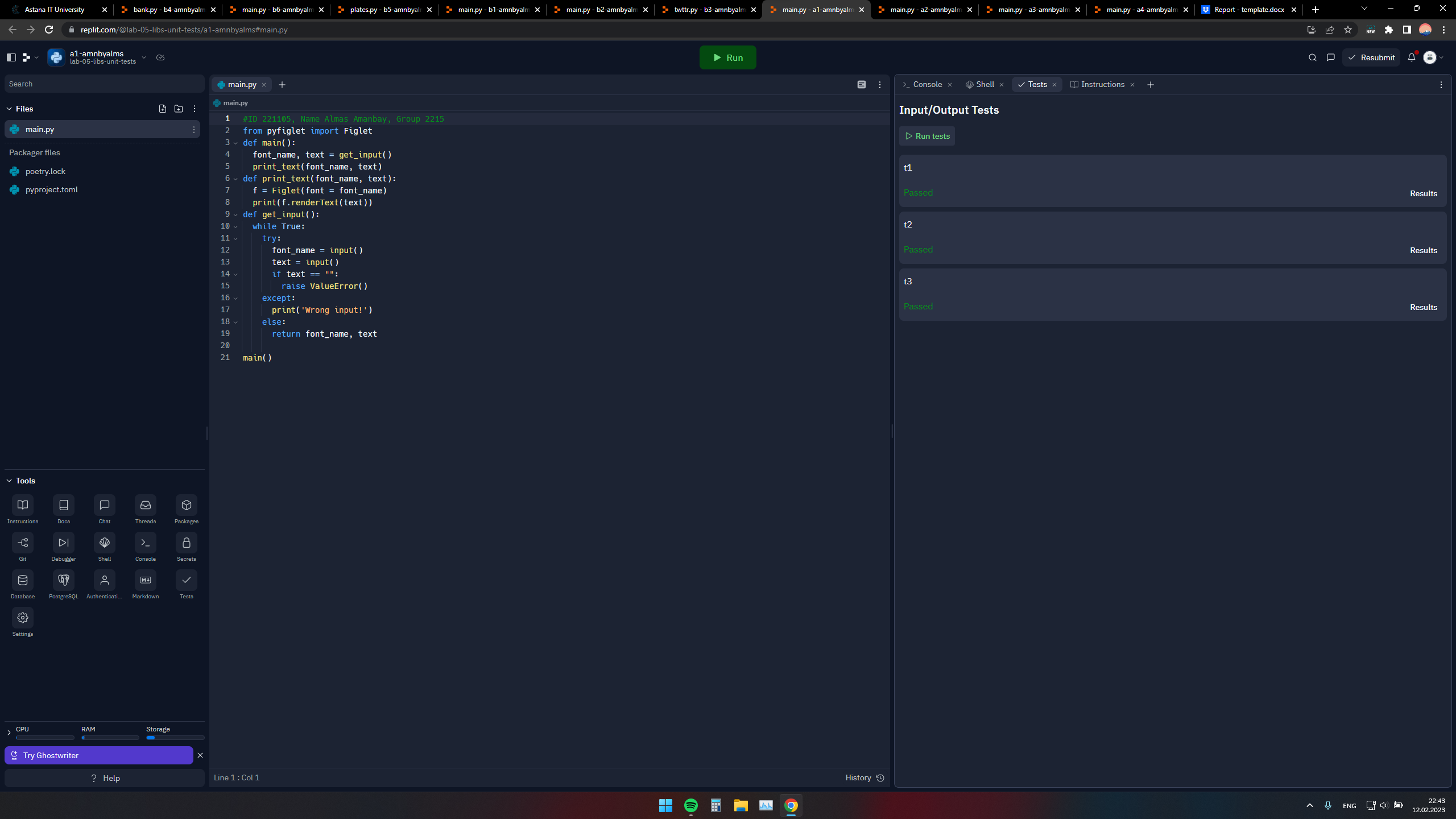
**except:**

**print('Wrong input!')**

**else:**

**return font\_name, text**

**main()**



**Problem -A2**

**Solution code**

**#Name Almas Amanbay, group 2215, ID 221105**

**from random import randint**

**def main():**

**play(get\_level())**

**def get\_level():**

**while True:**

**try:**

**level = int(input('Level:'))**

**if level < 1:**

**raise ValueError()**

**except:**

**pass**

**else:**

**return level**

**def play(level):**

**num = randint(1, level)**

**while True:**

**try:**

**guess = int(input('Guess:'))**

**if guess < 1:**

**raise ValueError()**

**except:**

**pass**

**else:**

**if guess < num:**

**print('Too small!')**

**elif guess > num:**

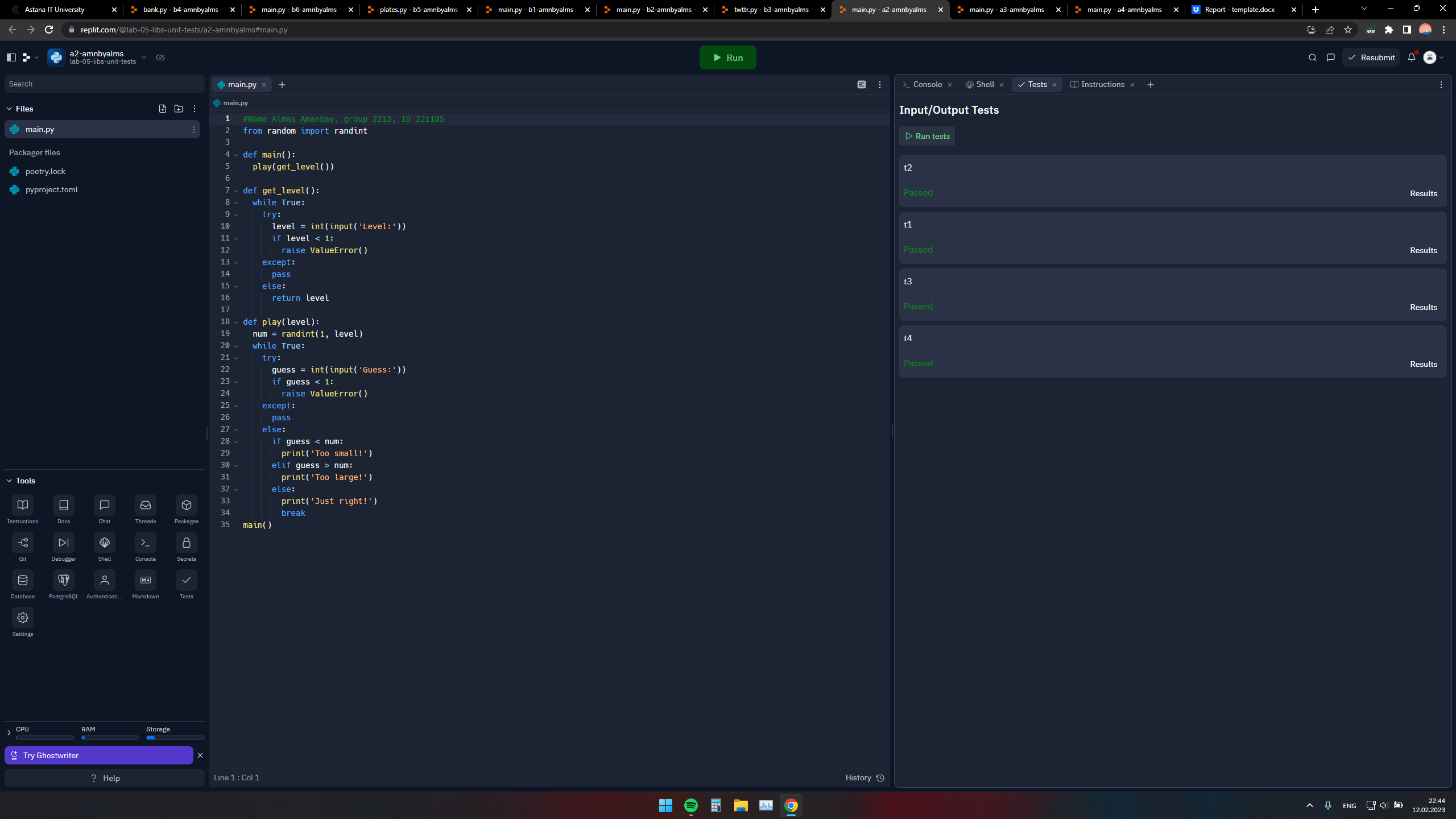
**print('Too large!')**

**else:**

**print('Just right!')**

**break**

**main()**



**Problem -A3**

**Solution code:**

#Name Almas Amanbay, Group 2215, ID 221105

def main():

A, B = get\_input()

op = int(input())

sol(A, B, op)

def get\_input():

A = set(input().split())

B = set(input().split())

return A, B

def union(A, B):

return A | B

def difference(A, B):

return A - B

def intersection(A, B):

return A & B

def symmetric\_Dif(A, B):

return A ^ B

def sol(A, B, op):

match op:

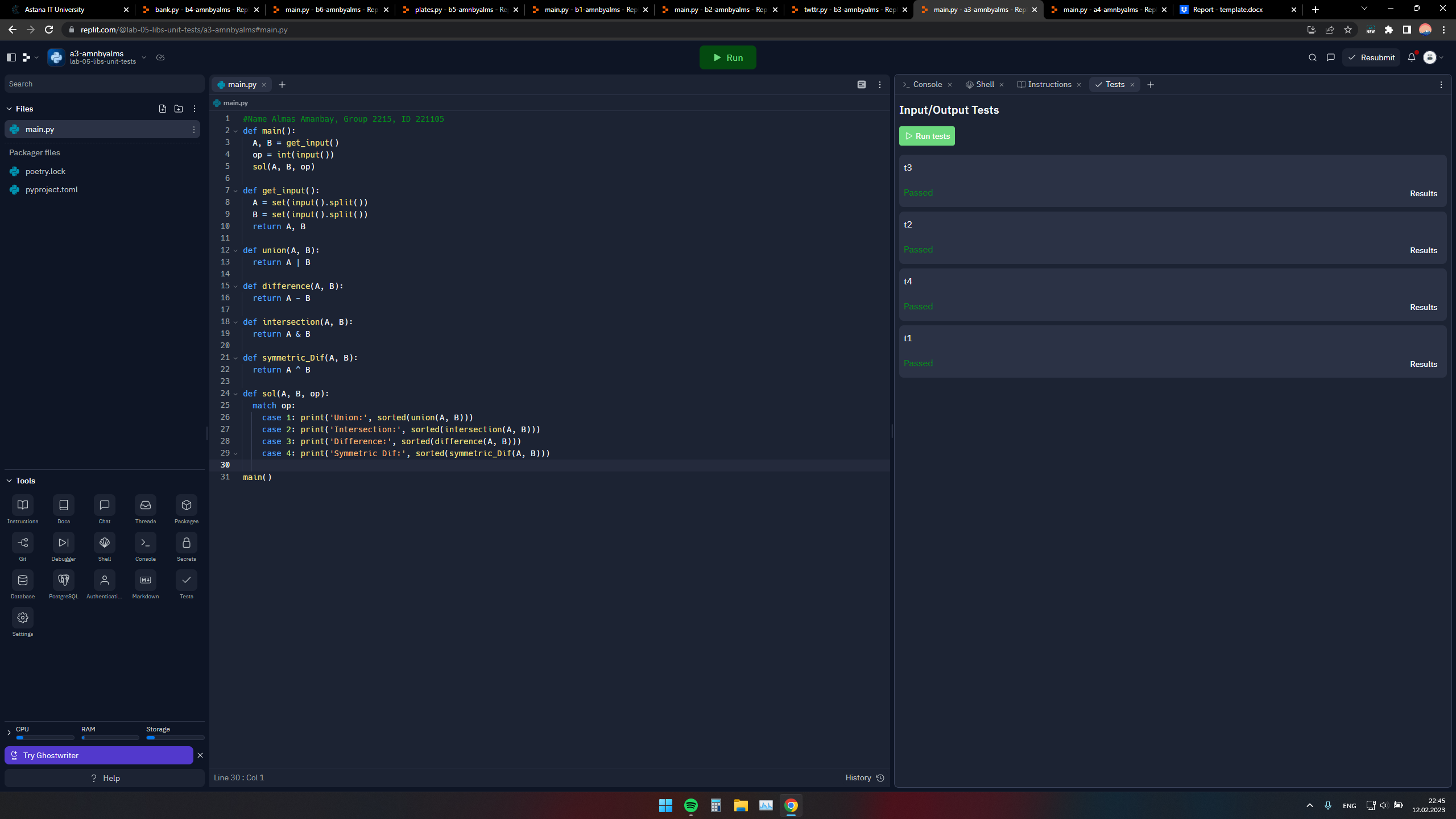
case 1: print('Union:', sorted(union(A, B)))

case 2: print('Intersection:', sorted(intersection(A, B)))

case 3: print('Difference:', sorted(difference(A, B)))

case 4: print('Symmetric Dif:', sorted(symmetric\_Dif(A, B)))

main()



**Problem-A4**

**Solution code:**

#Name Almas Amanbay, Group 2215, ID: 221105

def main():

t = get\_input()

print(sol(t))

def get\_input():

t1 = input().split()

t2 = input().split()

t = ()

for i in range(len(t1)):

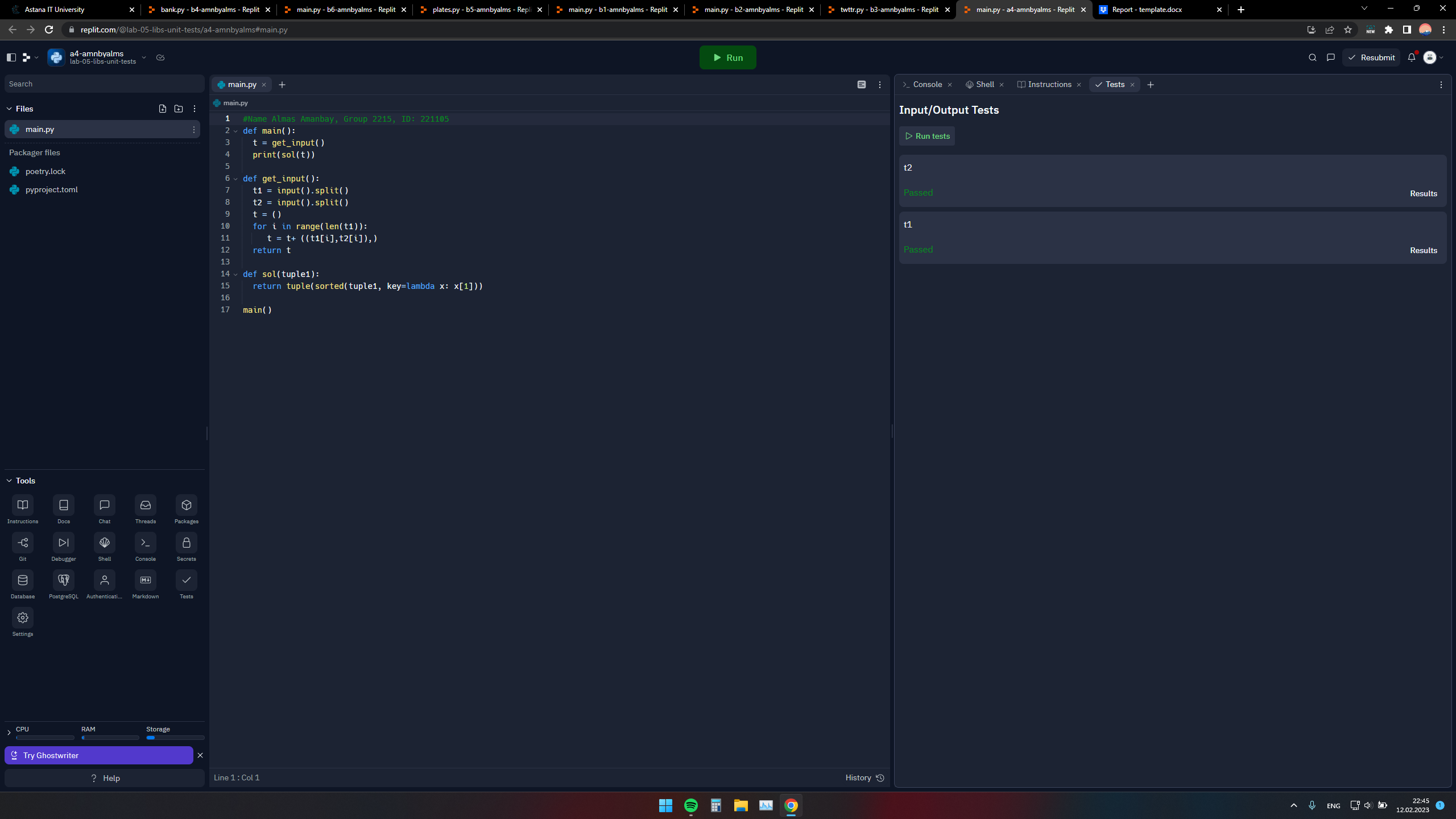
t = t+ ((t1[i],t2[i]),)

return t

def sol(tuple1):

return tuple(sorted(tuple1, key=lambda x: x[1]))

main()



**Problem-B1**

**Solution:**

#name Almas Amanbay,Group 2215, ID 221105

import emoji

import random

events = [':1st\_place\_medal:',':airplane:',':alien:',':bank:',

':beach\_with\_umbrella:',':house\_with\_garden:',

':couple\_with\_heart:',':baby:',':ewe:',':graduation\_cap:',':key:',

':laptop:',':money\_bag:',':money\_with\_wings:',':open\_book:',

':package:',':wine\_glass:',

':world\_map:',':gem\_stone:',':high-heeled\_shoe:',':tennis:']

deck = []

for \_ in range(3):

x = random.choice(events)

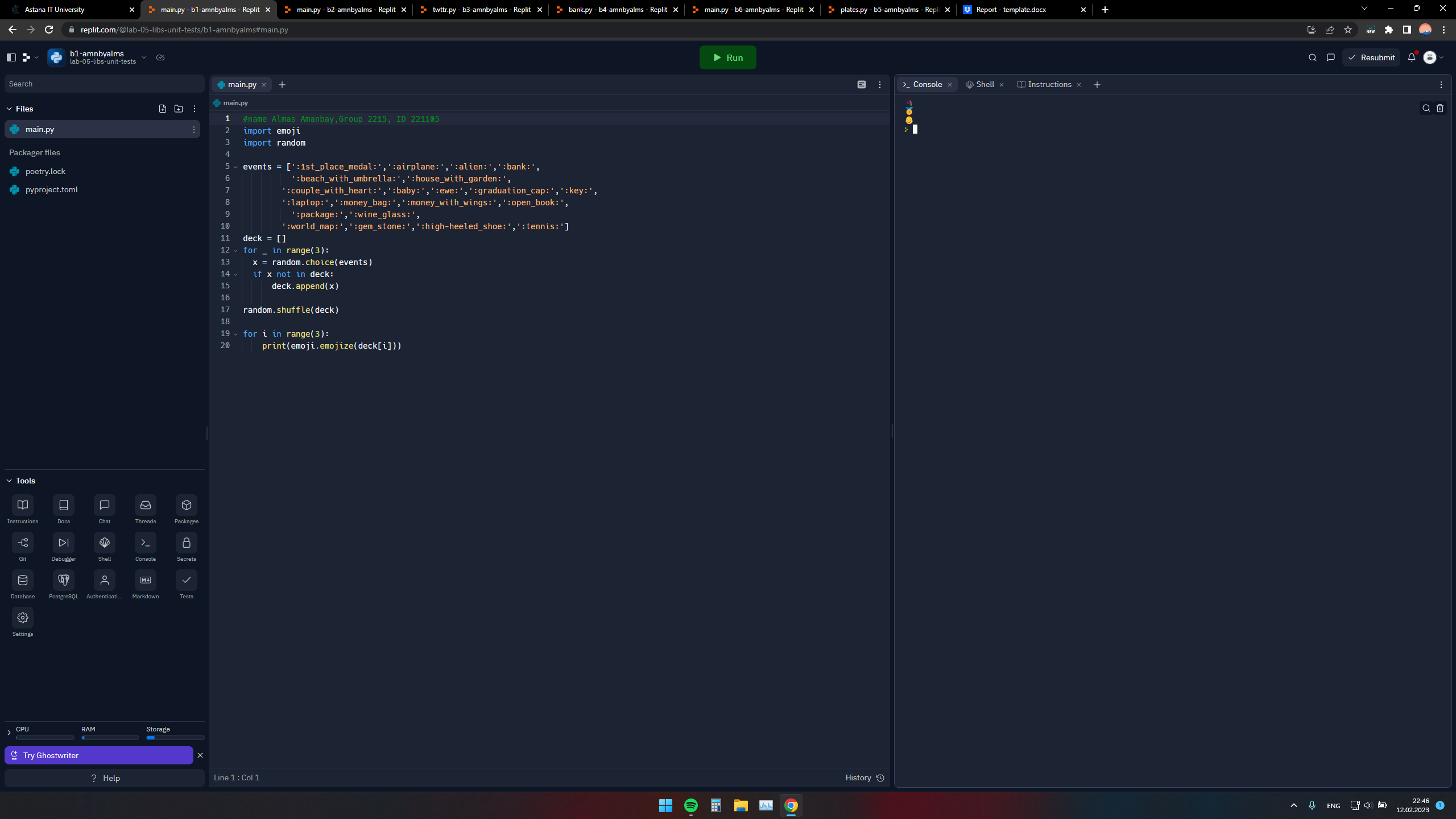
if x not in deck:

deck.append(x)

random.shuffle(deck)

for i in range(3):

print(emoji.emojize(deck[i]))



**Problem B2**

**Solution**

#name Almas Amanbay, Group 2215, ID 221105

from random import randint

def main():

generate\_integer(get\_level())

def get\_level():

while True:

try:

level = int(input('Level:'))

if level < 1 or level>3:

raise ValueError()

except:

pass

else:

return level\*2+3

def generate\_integer(level):

x=10

res=0

while x>0:

t=3

num1=randint(0,level)

num2=randint(0,level)

for i in range(3):

strin=(f'{num1} + {num2} =')

try:

equal = int(input(strin))

except ValueError:

print("EEE")

else:

if equal==num1+num2:

res+=1

break

else:

print("EEE")

t-=1

continue

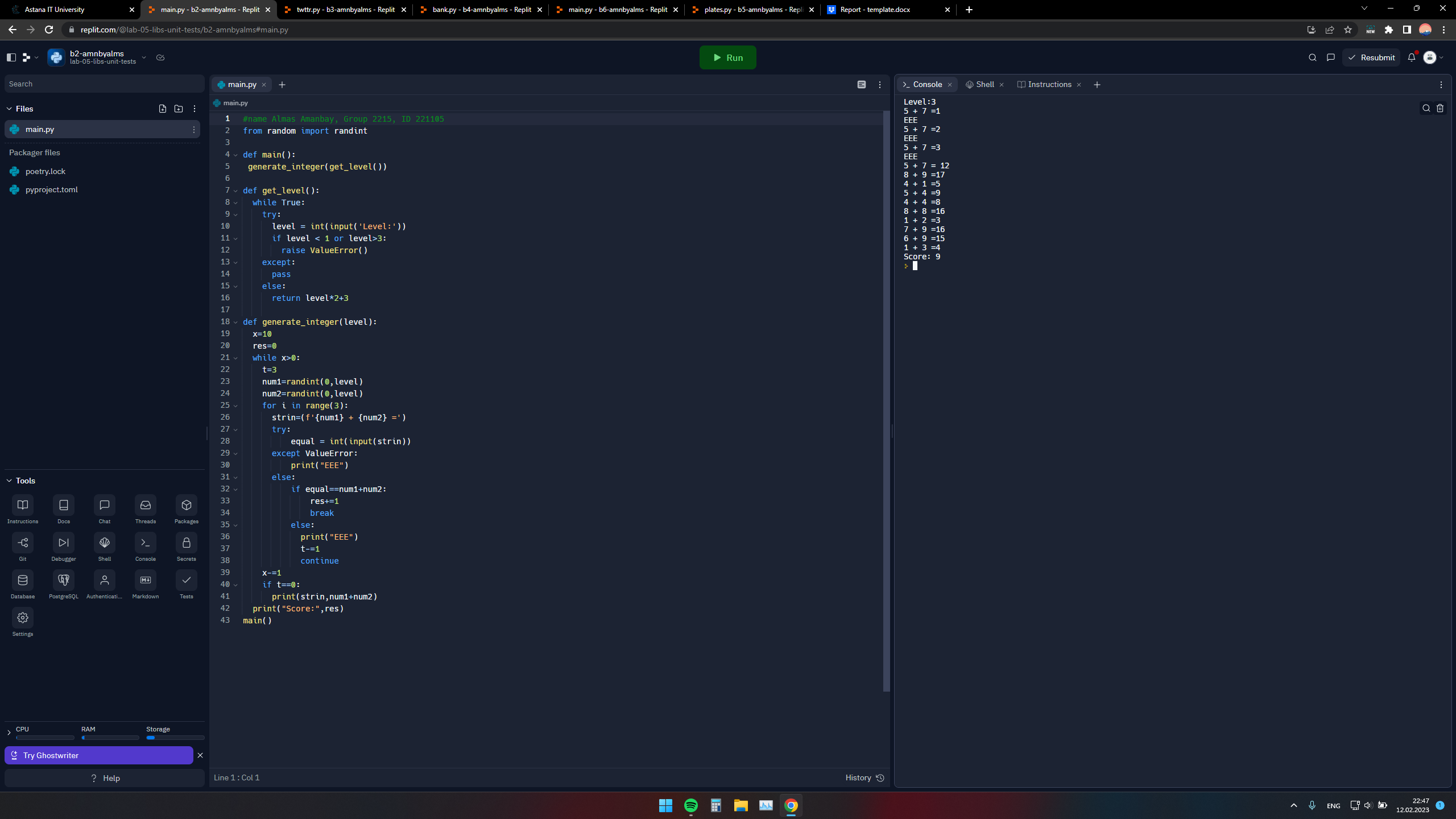
x-=1

if t==0:

print(strin,num1+num2)

print("Score:",res)

main()



**Problem B3**

**Twttr.py:**

#Name Almas Amanbay, Group 2215, ID 221105

def main():

text = input()

print(shorten(text))

def shorten(text):

for x in ['A','E','I','O','U','a','e','i','o','u']:

text = text.replace(x,'')

return text

if \_\_name\_\_ == "\_\_main\_\_":

main()

**test\_twtrr.py:**

from twttr import shorten

def test\_empty():

assert shorten("")==""

def test\_lowercase():

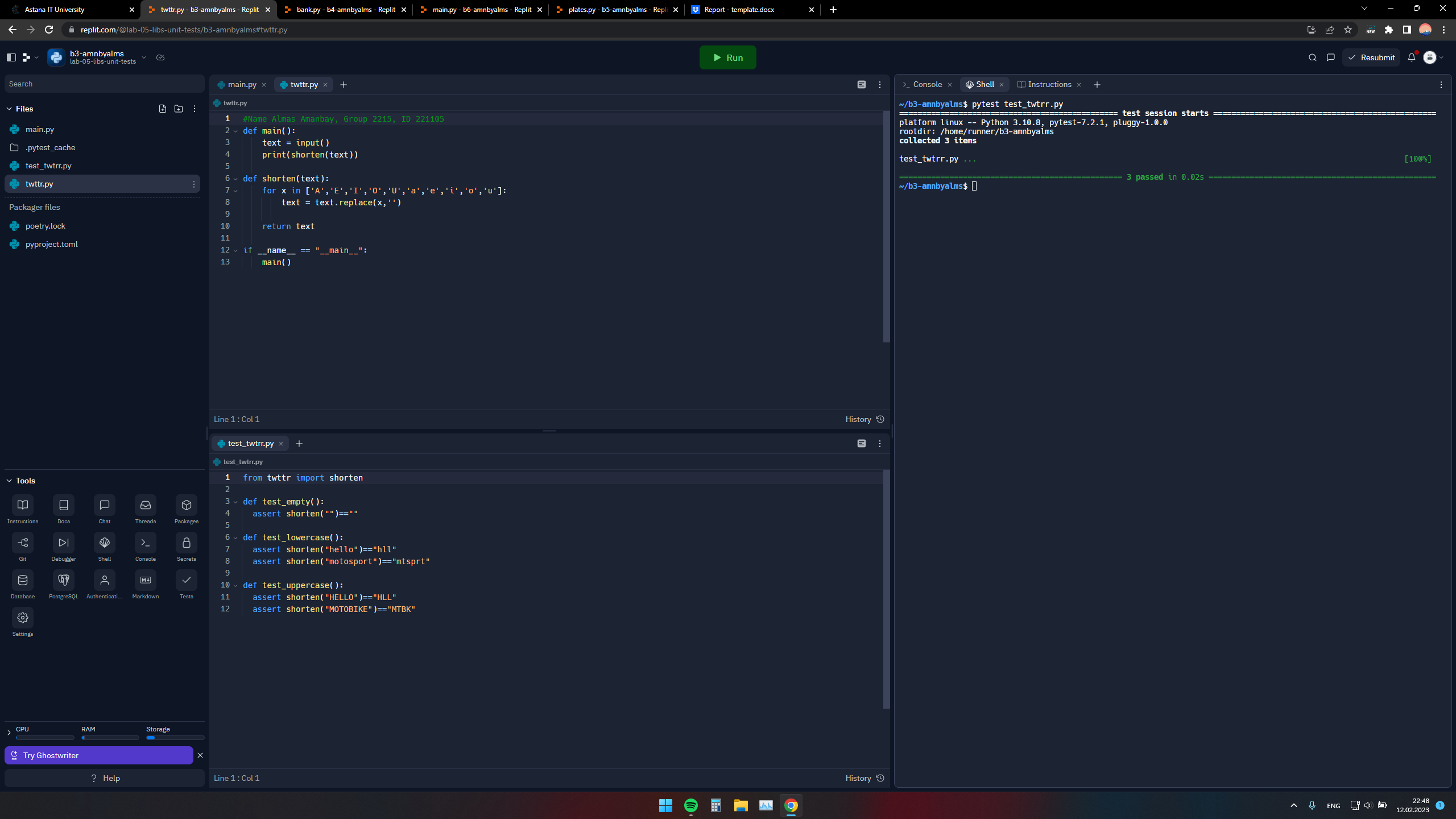
assert shorten("hello")=="hll"

assert shorten("motosport")=="mtsprt"

def test\_uppercase():

assert shorten("HELLO")=="HLL"

assert shorten("MOTOBIKE")=="MTBK"



**Problem B4**

**Solution**

**Bank.py**

#Name Almas Amanbay, Group 2215, ID 221105

def main():

greeting=input().lower().strip()

print(f"{value(greeting)}$")

def value(greeting):

greeting=greeting.lower()

if "hello" in greeting and greeting[0]=='h':

return 0

elif greeting[0]=='h':

return 20

else:

return 100

if \_\_name\_\_ == "\_\_main\_\_":

main()

**test\_bank.py:**

from bank import value

def test\_lowercase():

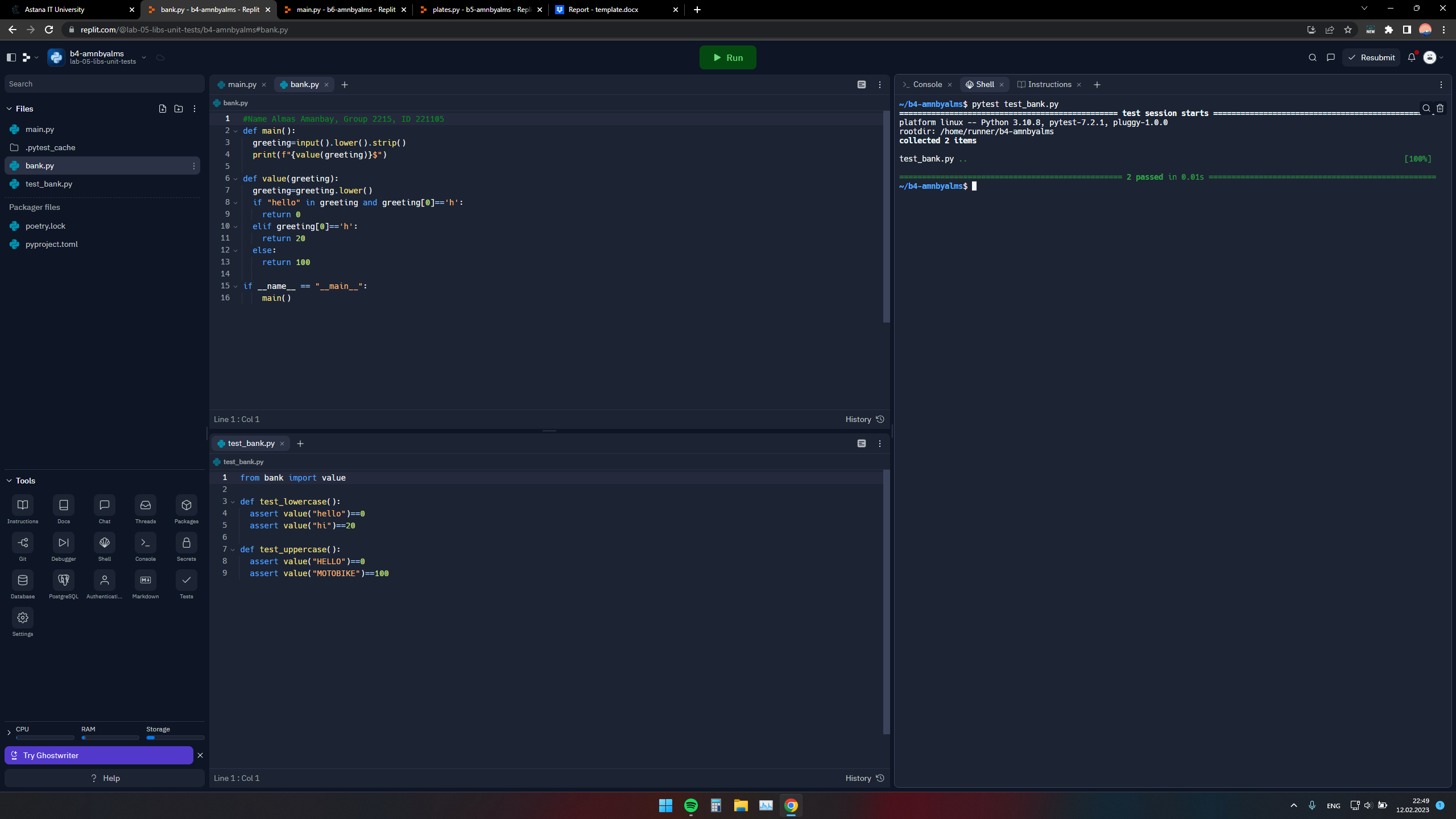
assert value("hello")==0

assert value("hi")==20

def test\_uppercase():

assert value("HELLO")==0

assert value("MOTOBIKE")==100



**Problem B5**

**Solution**

**Plates.py**

#Name Almas Amanbay, Group 2215, ID 221105

def main():

plate = list(map(str, input()))

if is\_valid(plate):

print("Valid")

else:

print("Invalid")

def is\_valid(s):

if (len(s) < 2 or len(s) > 6):return False

if len(s) % 2 == 0 and s[-1].isdigit() == False:

if s[int(len(s) / 2) - 1].isdigit():

return False

else:

if s[int(len(s) / 2)].isdigit() and s[-1].isdigit() ==False:

return True

for letter in s:

if letter.isdigit():

if (int(letter) == 0):

return False

break

return True

if \_\_name\_\_ == "\_\_main\_\_":

main()

**test\_plates.py:**

from plates import is\_valid

def test1():

assert is\_valid("SE2215") == True

def test2():

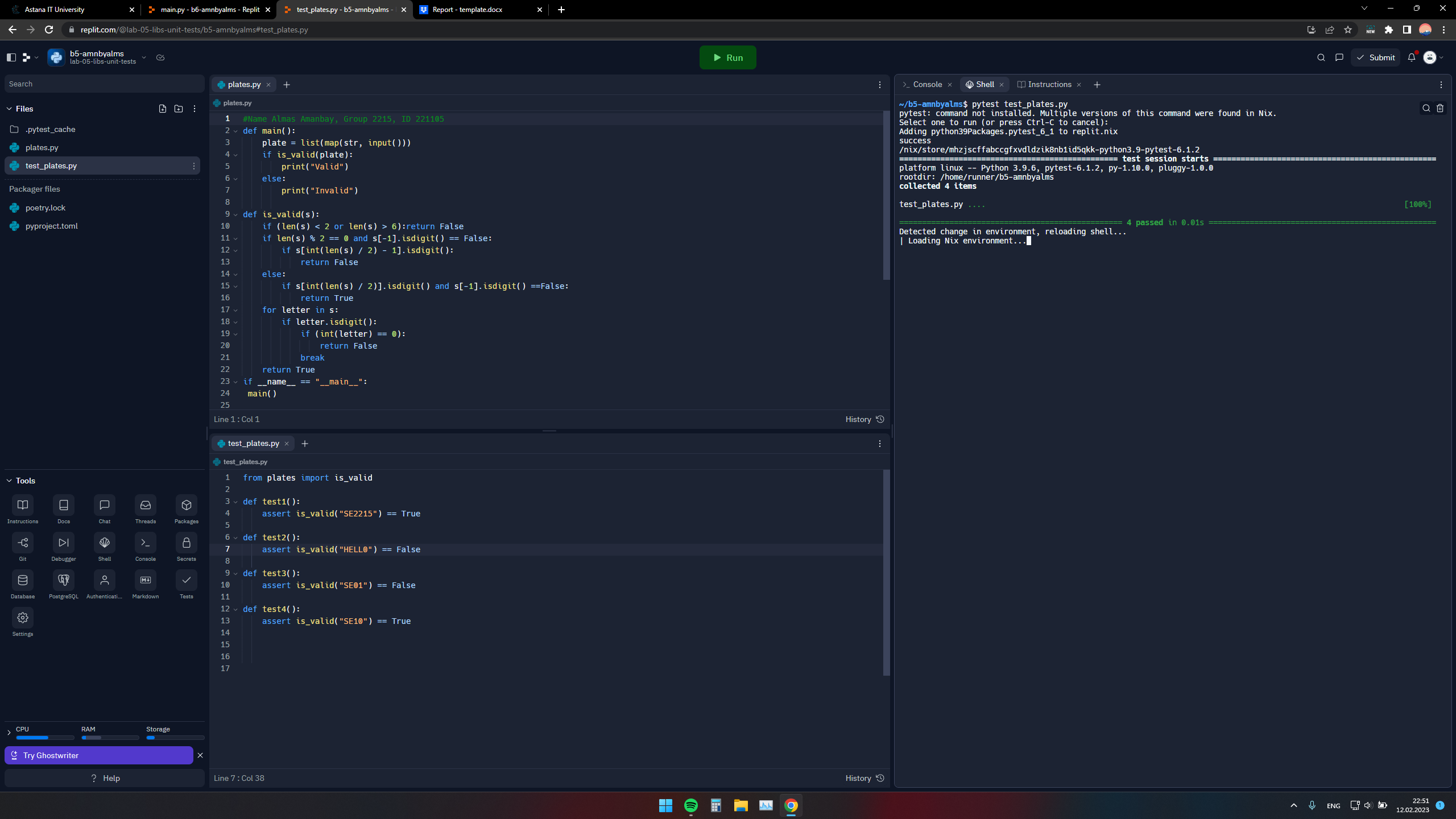
assert is\_valid("HELL0") == False

def test3():

assert is\_valid("SE01") == False

def test4():

assert is\_valid("SE10") == True



**Problem B6**

**Solution:**

**Fuel.py**

#Name Almas Amanbay, 221105, 2215

def main():

fraction = input()

print(convert(fraction))

def convert(fraction):

try:

if fraction.split('/')[0].isdigit() or fraction.split('/')[1].isdigit():

x = int(fraction.split('/')[0])

y = int(fraction.split('/')[1])

if x > y or y == 0:

raise ZeroDivisionError()

else:

fraction = int((x / y) \* 100)

return gauge(fraction)

else:

raise ValueError()

except ValueError:

print("Wrong input!")

except ZeroDivisionError:

print("Can't divide by zero or x is greater than y")

def gauge(percentage):

if percentage <= 1:

return "E"

elif percentage >= 99:

return "F"

else:

return f'{percentage}%'

if \_\_name\_\_ == "\_\_main\_\_":

main()

**test\_fuel.py**

#Name Almas Amanbay, group 2215, ID 221105

from fuel import convert

def test1():

assert convert('1/4') == '25%'

def test2():

assert convert('3/4') == '75%'

def test3():

assert convert('0/4') == 'E'

def test4():

assert convert('4/4') == 'F'

