**Q1.Consider a list (list = []). You can perform the following commands:  
insert i e: Insert integer  at position .  
print: Print the list.  
remove e: Delete the first occurrence of integer .  
append e: Insert integer  at the end of the list.  
sort: Sort the list.  
pop: Pop the last element from the list.  
reverse: Reverse the list.  
Initialize your list and read in the value of  followed by  lines of commands where each command will be of the  types listed above. Iterate through each command in order and perform the corresponding operation on your list.**

**Program:**

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

N = int(input()

lists = []

for i in range(N):

a = list(map(str,input().split( )))

lists.append(a)

arr = []

for x in lists:

if x[0] == "insert":

i = int(x[1])

e = int(x[2])

arr.insert(i,e)

elif x[0] == "print":

print(arr)

elif x[0] == "remove":

e = int(x[1])

arr.remove(e)

elif x[0] == "append":

e = int(x[1])

arr.append(e)

elif x[0] == "sort":

arr.sort()

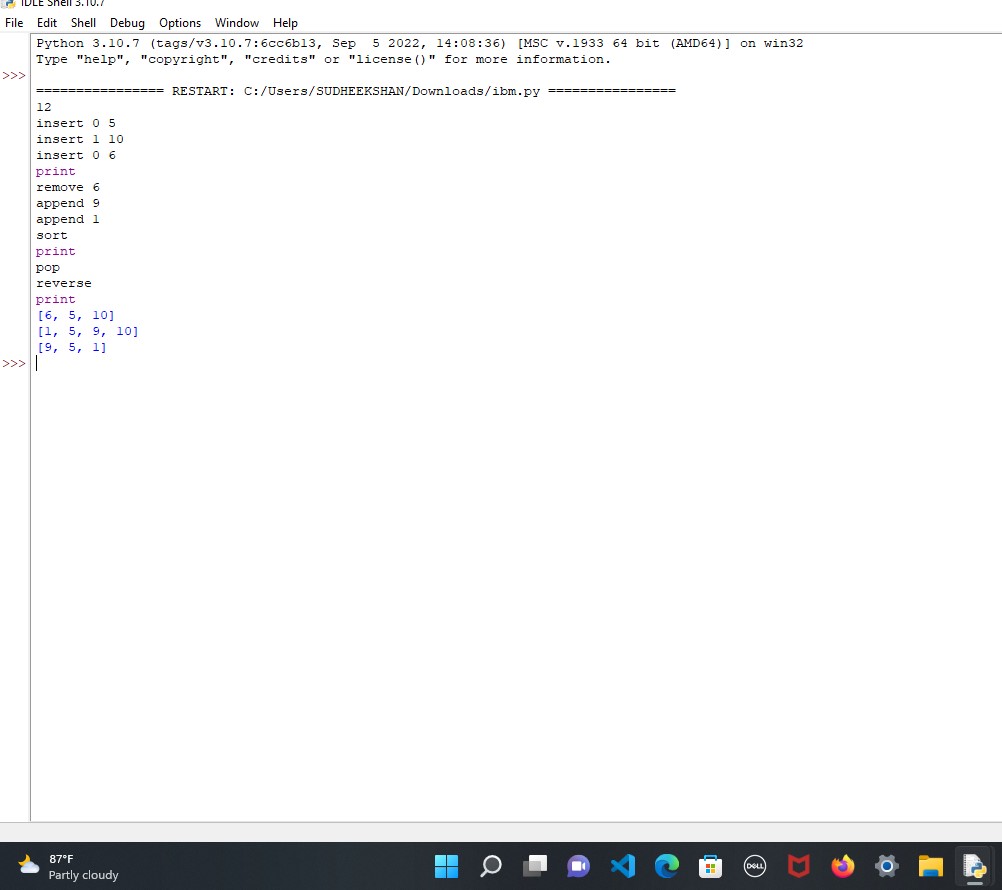
elif x[0] == "pop":

arr.pop()

elif x[0] == "reverse":

arr.reverse()

**Output**



**Q2. Write a Calculator program in Python?**

**Simple calcuator**

**Program:**

def add(P, Q):

   return P + Q

def subtract(P, Q):

   return P - Q

def multiply(P, Q):

   return P \* Q

def divide(P, Q):

   return P / Q

print ("Please select the operation.")

print ("a. Add")

print ("b. Subtract")

print ("c. Multiply")

print ("d. Divide")

choice = input("Please enter choice (a/ b/ c/ d): ")

num\_1 = int (input ("Please enter the first number: "))

num\_2 = int (input ("Please enter the second number: "))

if choice == 'a':

   print (num\_1, " + ", num\_2, " = ", add(num\_1, num\_2))

elif choice == 'b':

   print (num\_1, " - ", num\_2, " = ", subtract(num\_1, num\_2))

elif choice == 'c':

   print (num\_1, " \* ", num\_2, " = ", multiply(num\_1, num\_2))

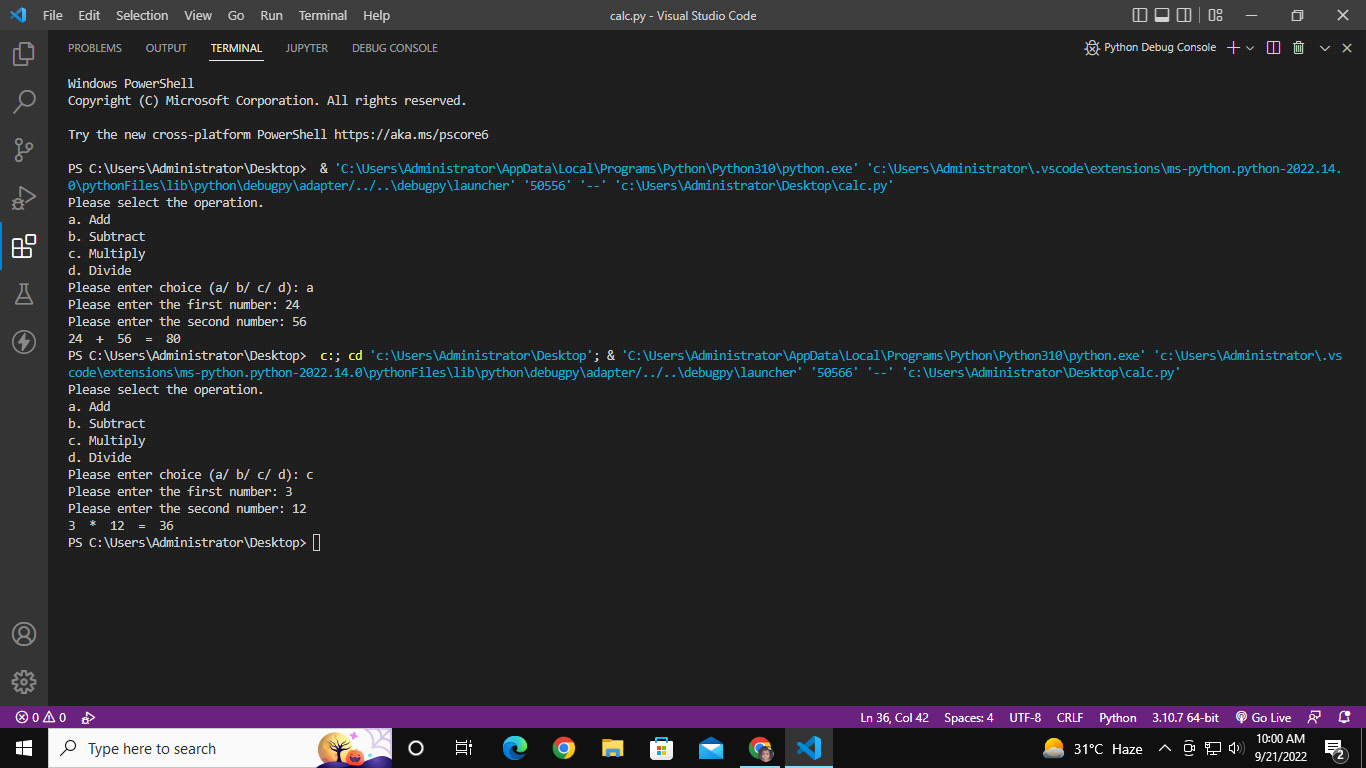
elif choice == 'd':

   print (num\_1, " / ", num\_2, " = ", divide(num\_1, num\_2))

else:

   print ("This is an invalid input")

**Output:**



**Q3. Write a program to concatenate, reverse and slice a string?**

**Program:**

def concat(x, y):

return x + y

def reverse(s):

str = ""

for i in s:

str = i + str

return str

def slicing(w, x, y):

num = 0

num = slice(x, y)

return w[num]

print("Select operation.")

print("1.Concatenate")

print("2.Reverse")

print("3.Slice")

while True:

choice = input("Enter choice(1/2/3): ")

if choice in ('1', '2', '3'):

if choice == '1':

str1 = input("Enter string1: ")

str2 = input("Enter string2: ")

print("After concatenation", concat(str1, str2))

elif choice == '2':

str3 = input("Enter string1: ")

print("After reversing", reverse(str3))

elif choice == '3':

str3 = input("Enter string1: ")

# num1 = input("Enter starting index: ")

#num2 = input("Enter stopping index: ")

# num3 = input("Enter increment: ")

print(str3[1:3])

next\_operation = input("Let's do next operation? (yes/no): ")

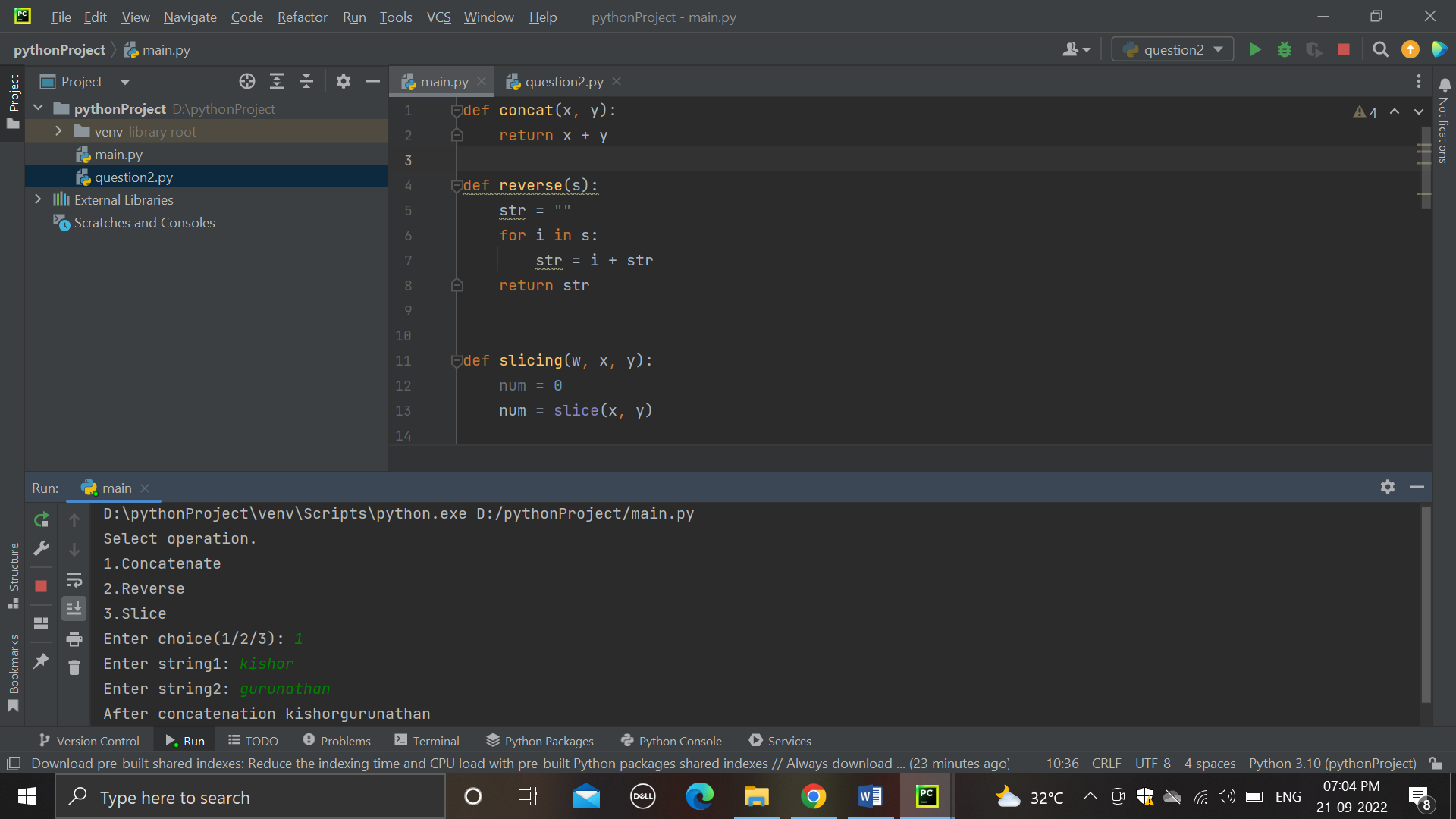
if next\_operation == "no":

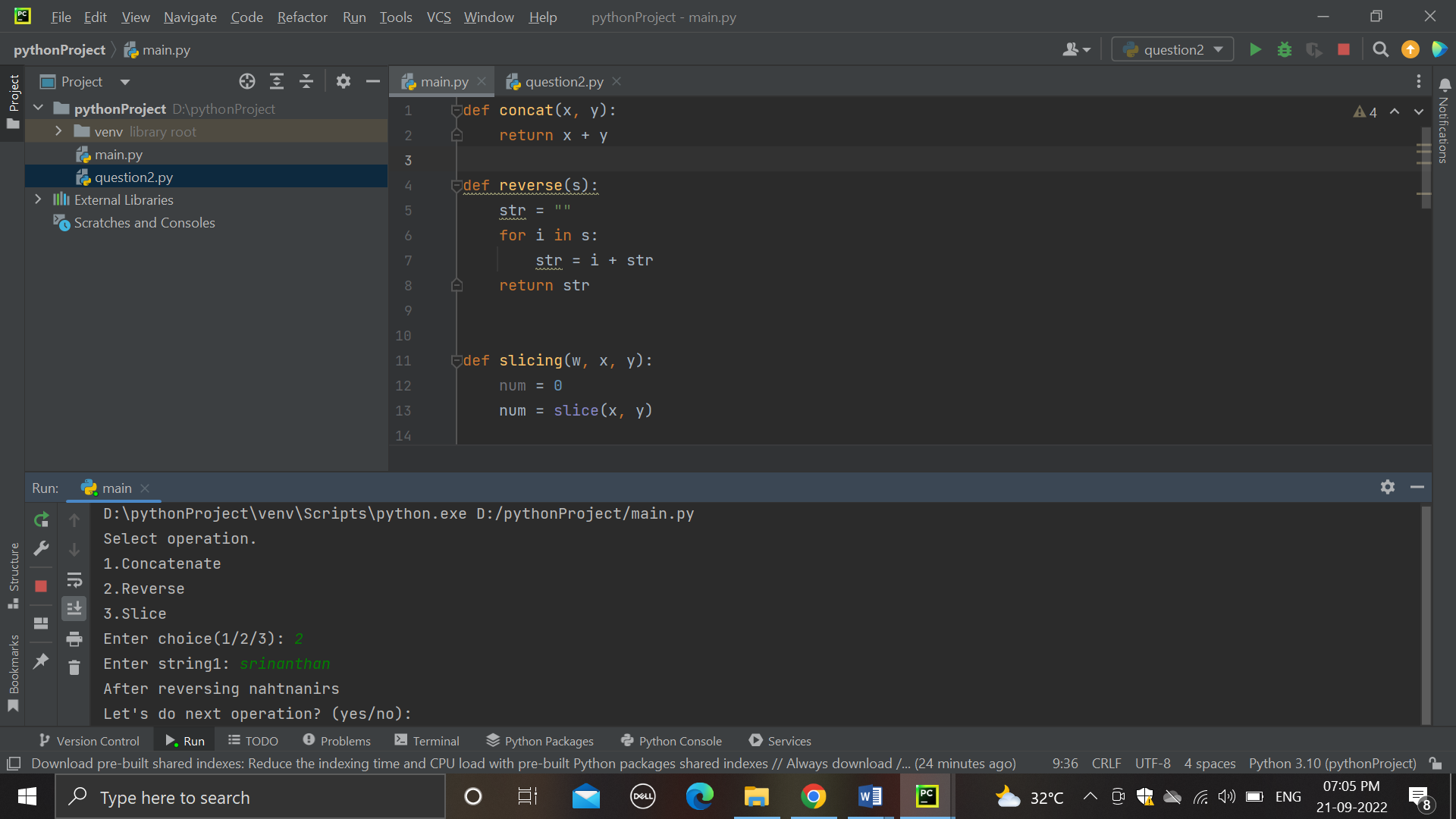
break

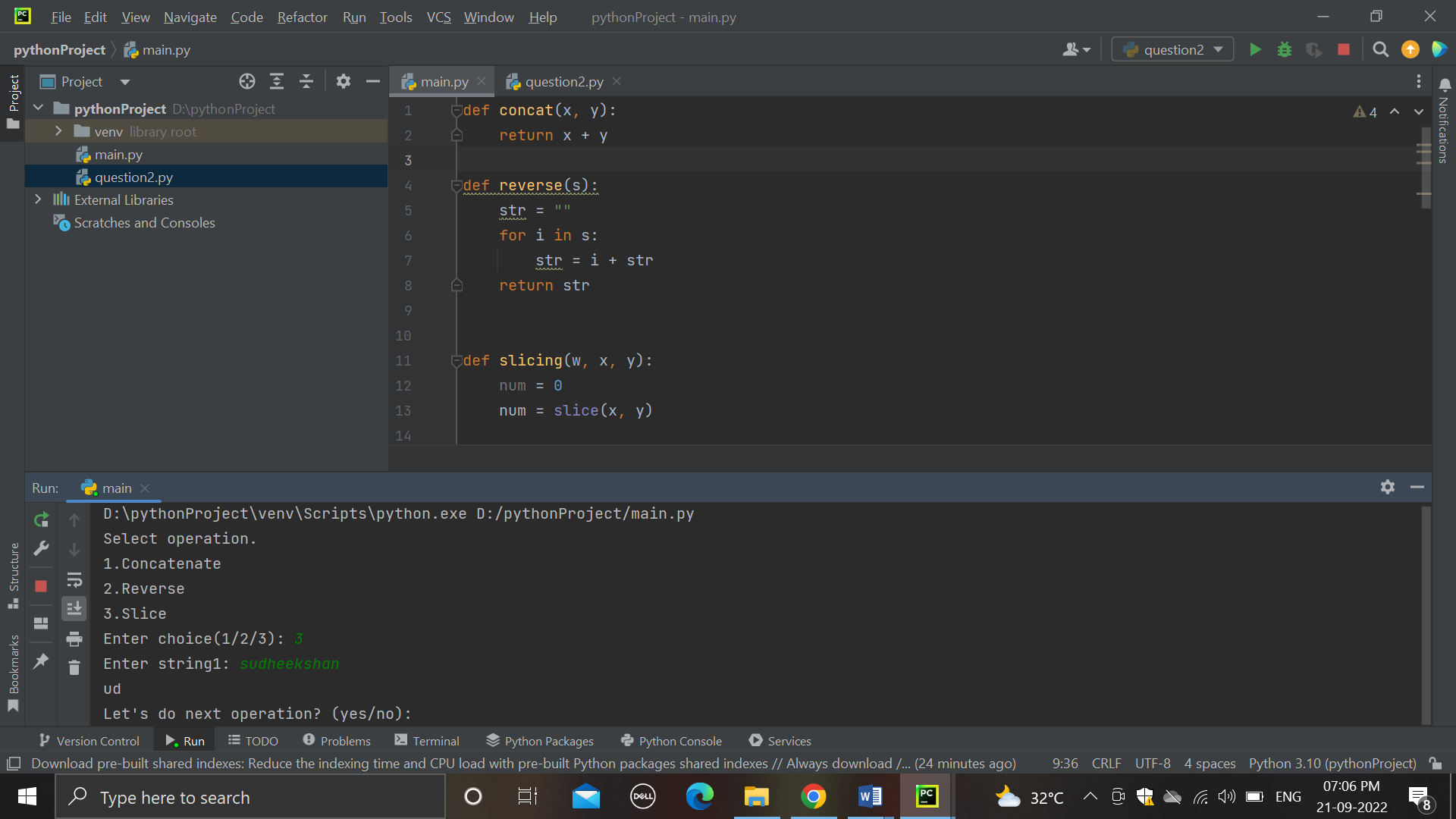
else:

print("Invalid Input")

**Output:**







**Q4. Why is Python a popular programming language?**

1. Python is easy to learn

2. Python has an active, supportive community

3. Python is flexible

4. Python offers versatile web-development solutions

5. Python is well suited to data science and analytics

6. Python is efficient, fast, and reliable

7. Python is widely used with IoT Technology

8. Python empowers custom automation

9. Python is the academic language

**Q5. What are the other Frameworks that can be used with python?**

1. Full-Stack Framework

A full-stack framework, also known as enterprise framework, is the one-stop solution for all development needs. These have built-in libraries configured to work seamlessly together. They support the development of databases, frontend interfaces, and backend services.

2. Microframework

Microframeworks are lightweight, minimalistic web application frameworks that have limited functionalities and features. Usually, microframeworks offer only those components that are required for building an application. They lack many additional functionalities and features like database abstraction layer, form validation, web template engine, authentication functionality, authorization, input validation, and input sanitation.

3. Asynchronous Framework

The asynchronous framework is the latest to join the Python framework bandwagon. It is a unique microframework that lets Developers handle and manage large sets of concurrent connections. These frameworks feed on Python’s Asyncio library.

**Q6. Full form of WSGI?**

The **Web Server Gateway Interfac**e (WSGI) is a standard interface between web server software and web applications written in Python.The Web Server Gateway Interface (WSGI, pronounced whiskey or WIZ-ghee) is a simple calling convention for web servers to forward requests to web applications or frameworks written in the Python programming language.