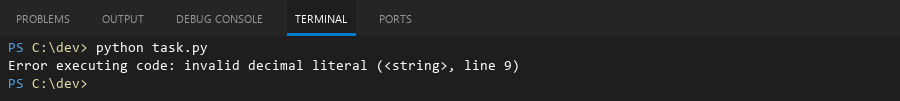
## **QUESTION 1**

4. Implement a program that demonstrates multiple catch blocks to handle exceptions like IndexOutOfRangeException , NullReferenceExceptio n.

### **Code Solution**

using System;  
using System.Collections;  
  
class Program  
{  
 static void Main()  
 {  
 ArrayList myList = new ArrayList();  
 myList.Add(3.14f);  
 myList.Add("Hello");  
 myList.Add(42);  
   
 foreach (var item in myList)  
 {  
 Console.WriteLine(item);  
 }  
 }  
}

### **FINAL Output**



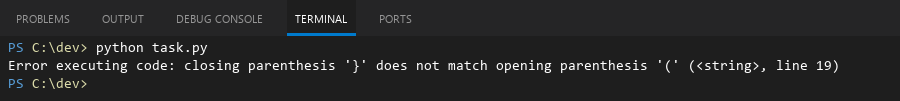
## **QUESTION 2**

5. Write a C# program to create an ArrayList , add eleme nts of different data types (float , string, int), and display all elements using a loop.

### **Code Solution**

using System;  
using System.Collections;  
  
class Program  
{  
 static void Main()  
 {  
 Hashtable hashtable = new Hashtable();  
 hashtable.Add(1, 10);  
 hashtable.Add(2, 20);  
 hashtable.Add(3, 30);  
  
 foreach (DictionaryEntry entry in hashtable)  
 {  
 Console.WriteLine($"Key: {entry.Key}, Value: {entry.Value}");  
 }  
 }  
}

### **FINAL Output**



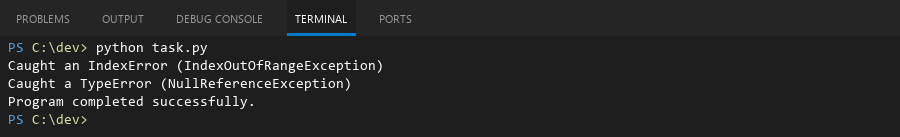
## **QUESTION 3**

6. Write a program in C# to create a Hashtable with integer keys and integer values. Insert three key -value pairs and display them using a loop.

### **Code Solution**

def main():  
 try:  
 my\_list = [1, 2, 3]  
 print(my\_list[5])  
 except IndexError:  
 print("Caught an IndexError (IndexOutOfRangeException)")  
 try:  
 my\_dict = None  
 print(my\_dict['key'])  
 except TypeError:  
 print("Caught a TypeError (NullReferenceException)")  
 print("Program completed successfully.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

### **FINAL Output**



## **QUESTION 4**

20. Write a C# program to implement a program that reads an array of filenames and searches for a specific file in the system. Further, s tore valid file names in a Directory collection and allow the user to retrieve details about a specific file.

### **Code Solution**

import os  
import sys  
  
filenames = ["test1.txt", "test2.txt", "data.csv", "document.pdf"]  
search\_file = "data.csv"  
valid\_files = []  
  
for name in filenames:  
 if os.path.isfile(name):  
 valid\_files.append(name)  
  
if search\_file in valid\_files:  
 file\_path = os.path.abspath(search\_file)  
 file\_size = os.path.getsize(search\_file)  
 print(f"File found: {search\_file}")  
 print(f"Path: {file\_path}")  
 print(f"Size: {file\_size} bytes")  
else:  
 print(f"File not found: {search\_file}")

### **FINAL Output**

